SYSTEMS, SCIENCE AND SOFTWARE

3SR-795

THE EFFECTS OF MESO-SCALE AND SMALL-SCALE INTERACTIONS ON GLOBAL CLIMATE

Semi-Annual Technical Report
for Period
15 February 1971 to 1 September 1971

Sponsored by:

Advanced Research Projects Agency
ARPA Order No. 1752
Program Code No. G1101D

Contract No.: DAHC 04-71-0018
Effective Date of Contract: 15 February 1971
Contract Expiration Date: 15 April 1972
Amount of Contract: $199,300.00

30 September 1971

P O. BOX 1620, LA JOLLA, CALIFORNIA 92037. TELEPHONE (714) 453-0060

3SR-795

THE EFFECTS OF MESO-SCALE AND SMALL-SCALE INTERACTIONS ON GLOBAL CLIMATE

Semi-Annual Technical Report
for Period
15 February 1971 to 1 September 1971

Sponsored by:

Advanced Research Projects Agency
ARPA Order No. 1752
Program Code No. G1101D

Contract No.: DAHC 04-71-0018
Effective Date of Contract: 15 February 1971
Contract Expiration Date: 15 April 1972
Amount of Contract: $199,300.00

30 September 1971

ABSTRACT

The present study was undertaken in an effort to improve numerical models for meso-scale and small-scale effects which influence global weather and its modification. Two major areas are being studied: the effects of mountain ranges on energy and momentum transfer, and the transient interaction of solar radiation with the earth's atmosphere. It is hoped that the results of these studies will lead to calculationally inexpensive prescriptions which can be incorporated into meso-scale and global-scale atmospheric circulation codes.

TABLE OF CONTENTS

TABLE OF CONTENTS, contd.

TABLE OF CONTENTS, contd.

NOMENCLATURE for SECTIONS 2-4

α = $\frac{u\Delta t}{\Delta x}$

C_p = specific heat at constant pressure

D = drag force on the obstacle

$\frac{d}{dt}$ = $\frac{\partial}{\partial t} + \vec{V}\cdot\nabla$

η = fluid vorticity = $\frac{\partial u}{\partial z} - \frac{\partial v}{\partial x}$

F = advective flux across a boundary

g = acceleration of gravity

Γ = dry adiabatic lapse rate = g/C_p

h = enthalpy

I = maximum value of the grid indice i

J = maximum value of the grid indice j

i,j = numerical grid indices

k_t = temperature diffusion constant

k_v = viscous diffusion constant

L = latent heat of vaporization for water

ℓ_c = cloud water content

ℓ_r = rain water content

P_r = water production terms

p = pressure

ϕ = compressibility stream function defined in Eq. (3.14)

ψ = stream function

Q = total water content

q = water contained as cloud moisture and vapor

R = gas constant for air

r = relative humidity

ρ = density

S = static stability

s = entropy

T = temperature

t = time

V_t = terminal velocity of water droplet in atmosphere

$\vec{V}$ = $iu + kw$ = total velocity

u = total horizontal velocity

w = vertical velocity

x = horizontal Cartesian coordinate

z = vertical Cartesian coordinate

ζ = compressibility vorticity function

$= \frac{\partial}{\partial z}(\rho u) - \frac{\partial}{\partial x}(\rho w)$

SUPERSCRIPTS

' = perturbation quantity

" = defined by Eq. (3.24)

n = time step index

SUBSCRIPTS

D = diameter of water droplet

i,j = numerical grid indices

o = initial spatial distribution (sometimes used to indicate a ground level value)

NOMENCLATURE for SECTION 5

ρ = density

P = pressure

T = temperature

z = vertical coordinate

ν = frequency

λ = wavelength

$\vec{\Omega}$ = unit vector

θ, ϕ = spherical angular coordinates defining $\vec{\Omega}$

μ = $\cos\theta$

I_ν = specific intensity of radiation

J_ν = radiation source function

κ_ν = volume extinction coefficient

α_ν = volume absorption coefficient

α'_ν = α_ν corrected for stimulated emission

β_ν = volume scattering coefficient

B_ν = Planck function

P_ν = scattering phase function

h = Planck's constant

k = Boltzmann's constant

θ_s = scattering angle

μ_s = $\cos\theta_s$

NOMENCLATURE for SECTION 5, contd.

$\bar{I}_\nu$ = azimuthal average of I_ν

$\bar{P}_\nu$ = azimuthal average of P_ν

E_ν = radiation energy density

$F_{\alpha,\nu}$ = α-component of radiation energy flux

$P_{\alpha\beta,\nu}$ = $\alpha\beta$-component of radiation pressure tensor

C_p = specific heat of air at constant pressure

F = frequency-integrated vertical flux

$\beta_{\nu,R}$ = Rayleigh volume scattering coefficient

$\beta_{\nu,M}$ = Mie volume scattering coefficient

P_R = Rayleigh phase function

$P_{\nu,M}$ = Mie phase function

n_s = index of refraction of air at 760 mm Hg and 15°C

N = number density of air molecules

N_s = number density of air molecules at 760 mm Hg and 15°C

ρ_n = depolarization factor for Rayleigh scattering

i_1, i_2 = Mie scattering functions

a = radius of (spherical) aerosol particle

m = complex index of refraction

$n(a)$ = probability distribution of aerosol radii

NOMENCLATURE for SECTION 5, contd.

N_{aer} = number density of aerosols

σ_{ext} = extinction cross section of a spherical particle

σ_{sca} = scattering cross section of a spherical particle

σ_{abs} = absorption cross section of a spherical particle

S_ν = solar intensity

A_ν = directional-hemispherical reflectivity

ρ_ν = bidirectional reflectivity

$\bar{\rho}_\nu$ = azimuthal average of ρ_ν

ε_ν = directional emissivity

T_g = ground (surface) temperature

$\bar{I}_\nu^{solar}$ = solar-beam part of $\bar{I}_\nu$

$\bar{I}_\nu^{diff}$ = diffuse part of $\bar{I}_\nu$

I_i = frequency-averaged $\bar{I}_\nu$

$\bar{\nu}_i$ = $\frac{1}{2}(\nu_i + \nu_{i+1})$ = center of frequency interval (ν_i, ν_{i+1})

β_i, B_i, P_i = β_ν, B_ν, P_ν evaluated at $\bar{\nu}_i$

T_i = transmission function

$\hat{S}_\nu$ = scattering source

$\hat{S}_i$ = frequency-averaged scattering source

$q^{(\alpha)}_{ijk\ n}$ = moments of $P_{i\ xi}$

1. INTRODUCTION

The numerical prediction of the general circulation of the atmosphere predates most of the other applications of high-speed computers to physical problems. The codes which exist at several major research centers have reached levels of considerable sophistication. These codes are used to solve time-dependent equations describing atmospheric motion in a three-dimensional representation. Parametric descriptions are included to take into account the effects of insolation, turbulent transport, and moisture. For the application to short period forecasts (covering a time interval of several days), the physical processes taken into account in the codes are quite satisfactory, relying on the kinetic and internal energy already in the atmosphere and depending less on the utilization of the energy source from insolation.

For predictions over a longer period of time the processes which transform the solar energy into motion of the atmosphere are much more important. The many phenomena which affect transfer of energy and moisture through the earth-ocean-atmosphere system are incompletely described. Descriptions of the ocean-air, air-land, and land-ocean interfaces, and of the topographic boundary conditions are necessary for a qualitatively correct predictive model.

1.1 OROGRAPHIC EFFECTS ON GLOBAL CLIMATE

Phenomena taking place on a scale smaller than the resolution of global circulation codes can cause changes in

climate. The tropospheric transport coefficients that are required in the global atmospheric model may arise from atmospheric motions that occur in quite small regions (e.g., mountain lee waves). Transport is also effected by convective eddies such as cumulus and cumulo-nimbus convective cells. These may be influenced by small geographic features such as islands and by upper atmospheric phenomena such as jet streams and waves.

The simplest method of accounting for meso-scale phenomena is to calculate parameters (such as eddy diffusivities) according to some fit to experimental data, risking large inaccuracies due to incomplete and inappropriate data. A technique which can give more accuracy is to compute these parameters by means of several meso-scale calculations performed separately, or concurrently with the large scale calculation. This permits a more complete description of relevant physical processes to be built into the global model.

Present research at Systems, Science and Software (S^3) concerns the development of a meso-scale code capable of studying these phenomena and in presenting calculational results which may be incorporated into global scale codes. The basic code, as discussed in this report, is a two-dimensional time-dependent code which makes use of the Boussinesq approximation. The code is described in Section 2. Several test calculations have been completed which show the transient effects on the air flow over mountain ranges under various atmospheric conditions. These results are described in Section 4. The momentum transport from the atmosphere to the earth is calculated for two cases and these results are also discussed.

Modifications to the code, reported in Section 3, include the effects of moisture, variable zoning in the

vertical direction in order to better describe the atmospheric conditions (i.e., inversion layers, etc.) and consideration of the compressibility effects of the atmosphere. The results of test calculations using these new codes will be reported in the final report of this contract.

1.2 RADIATIVE TRANSFER IN CLIMATOLOGY

To quantify the sources and sinks of energy in the atmosphere due to solar and terrestrial radiation as a function of location, season, and time is a central problem in predictive climatology. Radiation is the source which strongly influences the level of response for all other parts of the system. A number of parameters depend sensitively on the solar radiation: humidity, cloudiness, extent of snow and ice, etc., and, in turn, the amount of solar radiation heating the air and land depends on them.

Because of the intrinsic difficulty of the radiative transfer calculation, very substantial approximations have been made in the descriptions of radiative effects in all atmospheric codes. Most calculations of atmospheric radiation have been limited to approximations of long-wave cooling. Only a few transient calculations have been performed and the radiative response of the lower boundary of the atmosphere has been very crudely approximated or ignored.

The development to date of a one-dimensional radiative transfer code which will take into account the time-dependent modifications in the thermal stratification of the atmosphere is described in Section 5. A one-dimensional boundary layer code which describes the basic hydrodynamics of the air flow and which will be used to test the radiative transfer code is also described. The radiative transfer code will be capable of characterizing the transfer through an atmosphere described

by temperature, pressure, humidity, CO_2 concentration, O_3 concentration, and concentrations of other trace constituents including aerosols.

In summary, the two major areas under investigation are (1) the effects of mountain ranges on energy and momentum transfer, and (2) the transient interaction of solar radiation with the earth's atmosphere. The development of numerical models to study these phenomena is described in Sections 2, 3, and 5 of this report. A sixth section, describing additional objectives to be pursued in connection with these investigations is also included.

2. OROGRAPHIC EFFECTS

The effects of mountain ranges on atmospheric transport is being investigated using a two-dimensional numerical code HAIFA (Hydrodynamics in an Almost Incompressible Flow Approximation). This code calculates time-dependent dynamic flow based on the Boussinesq approximation. The description of the basic code is contained in this section. The modifications to this code to incorporate the effects of moisture and compressibility are discussed in Section 3.

2.1 THE HAIFA EQUATIONS

The numerical investigation of mountain waves requires that the effects of inertia and buoyancy be taken into account. The two-dimensional time-dependent Boussinesq equations, developed herein, include these effects in the HAIFA computer code. The buoyancy effects are due to adiabatic changes of temperature induced by perturbations of an initially thermally stratified atmosphere. Deviations from constancy of the density in other terms of the fluid equations, including the continuity equation, are neglected, giving a set of equations which are basically valid for an incompressible fluid. The use of the Boussinesq equations for the investigation of mountain waves, therefore, is appropriate in that the effects of buoyant stability is restricted by the incompressibility of the flow. These equations, as used in HAIFA, are the vorticity equation derived from the two-dimensional equations of motion, the

energy equation, and the continuity equation for an incompressible fluid. An outline of the derivation of these equations follows. (The symbols used in the equations are defined in the Nomenclature list.)

In the Boussinesq approximation, the momentum equations in the horizontal (x) and the vertical (z) directions are:

$$\frac{du}{dt} = -\frac{1}{\rho_o}\frac{\partial p}{\partial x} + \nabla\cdot(k_v \nabla u) \quad , \tag{2.1}$$

$$\frac{dw}{dt} = -\frac{1}{\rho_o}\frac{\partial p}{\partial z} - \frac{\rho}{\rho_o} g + \nabla\cdot(k_v \nabla w) \quad . \tag{2.2}$$

For the present, we have neglected the Coriolis terms in this set of equations.

The incompressible continuity equation in two dimensions is

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \quad . \tag{2.3}$$

The vorticity equation used in the HAIFA code is derived using Eqs. (2.1), (2.2), and (2.3). Eq. (2.1) is differentiated with respect to z and Eq. (2.3) with respect to x. Consistent with the Boussinesq approximation, the variation of ρ_o with height is assumed negligible. Subtracting one from the other removes the pressure terms. If one also treats the diffusion coefficient k_v as a constant, the resulting expression is:

$$\frac{d}{dt}(\eta) = +\frac{g}{\rho_o}\frac{\partial \rho}{\partial x} + k_v \nabla^2(\eta) \quad , \tag{2.4}$$

where η is defined as the vorticity component perpendicular to the x-z plane. Mathematically,

$$\eta = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \quad .$$

It is further possible to modify Eq. (2.4) consistent with the Boussinesq approximations. The variables ρ, T and p may be written as functions of their static values plus a perturbation contribution as follows:

$$\begin{aligned} \rho(x,z,t) &= \rho_o(z) + \rho'(x,z,t) \quad , \\ T(x,z,t) &= T_o(z) + T'(x,z,t) \quad , \\ p(x,z,t) &= \rho_o(z) + \rho'(x,z,t) \quad . \end{aligned} \tag{2.5}$$

The buoyancy term

$$\frac{1}{\rho_o} \frac{\partial \rho}{\partial x}$$

can then be written as

$$\frac{1}{\rho_o} \frac{\partial \rho'}{\partial x} \quad .$$

However, for the Boussinesq approximation to be valid, the density variation ρ' must depend mainly on temperature, i.e., the variation of density due to the dynamical pressure is assumed negligible (see Appendix A). Therefore,

$$\rho' = \left(\frac{\partial \rho}{\partial T}\right)_p T' = - \frac{\rho_o}{T_o} T' \quad . \tag{2.6}$$

Substituting Eq. (2.6) into Eq. (2.4) and using Eq. (2.3) to allow the result to be written in conservative form, the vorticity equation is

$$\frac{\partial}{\partial t}(\eta) + \frac{\partial}{\partial x}(u\eta) + \frac{\partial}{\partial z}(w\eta) = - \frac{g}{T_o} \frac{\partial T'}{\partial x} + k_v \nabla^2 \eta \quad . \tag{2.7}$$

Eq. (2.7) is the first of three equations to be solved in the HAIFA code. The second equation results from the continuity equation and the definition of vorticity. Defining a stream function ψ such that $u = \partial\psi/\partial z$ and $w = -\partial\psi/\partial x$, the continuity equation is automatically satisfied. Further, the stream function is related to the vorticity through a Poisson equation of the form

$$\nabla^2 \psi = \eta \quad . \tag{2.8}$$

The final equation necessary to complete the description of mountain waves is the energy equation. This equation expresses the first law of thermodynamics

$$\frac{dh}{dt} = + \frac{1}{\rho} \frac{dp}{dt} + k_t \nabla^2 h$$

for an adiabatic system. For a perfect gas with constant specific heat and using the hydrostatic approximation in the dp/dt term, this equation may be expressed by

$$\frac{dT}{dt} = - \frac{1}{C_p} wg + k_t \nabla^2 T \quad . \tag{2.9}$$

Substituting Eq. (2.5) into Eq. (2.9), the resulting energy equation is

$$\frac{\partial T'}{\partial t} + \frac{\partial}{\partial x}(uT') + \frac{\partial}{\partial z}(wT') = -w \frac{\partial T_o}{\partial z} + \Gamma + k_t \nabla^2 T' \quad . \qquad (2.10)$$

Eqs. (2.7), (2.8) and (2.10) constitute the fluid flow equations integrated in the HAIFA code.

2.2 NUMERICAL APPROXIMATION OF HAIFA EQUATIONS

Eqs. (2.7), (2.8) and (2.10) are written in finite difference form and integrated numerically. The integration is accomplished by updating the equations in time for each variable based on the values at the previous time step or an intermediate time located between two successive time steps. Each of these steps will be discussed in turn in this report. These descriptions include the definition of the grid used and the location of each variable listed in the equations, the evaluation of the advection terms in the vorticity and energy equations, the solution for the stream function from the Poisson eq. (2.8), and a discussion of the boundary conditions used in the numerical integration.

2.2.1 Finite Difference Scheme

The basic scheme used to numerically integrate the HAIFA equations is shown in Figure 2.1. The finite difference grid used in HAIFA is shown in Figure 2.2. The locations of the major variables with respect to the grid cells are defined in the figure.

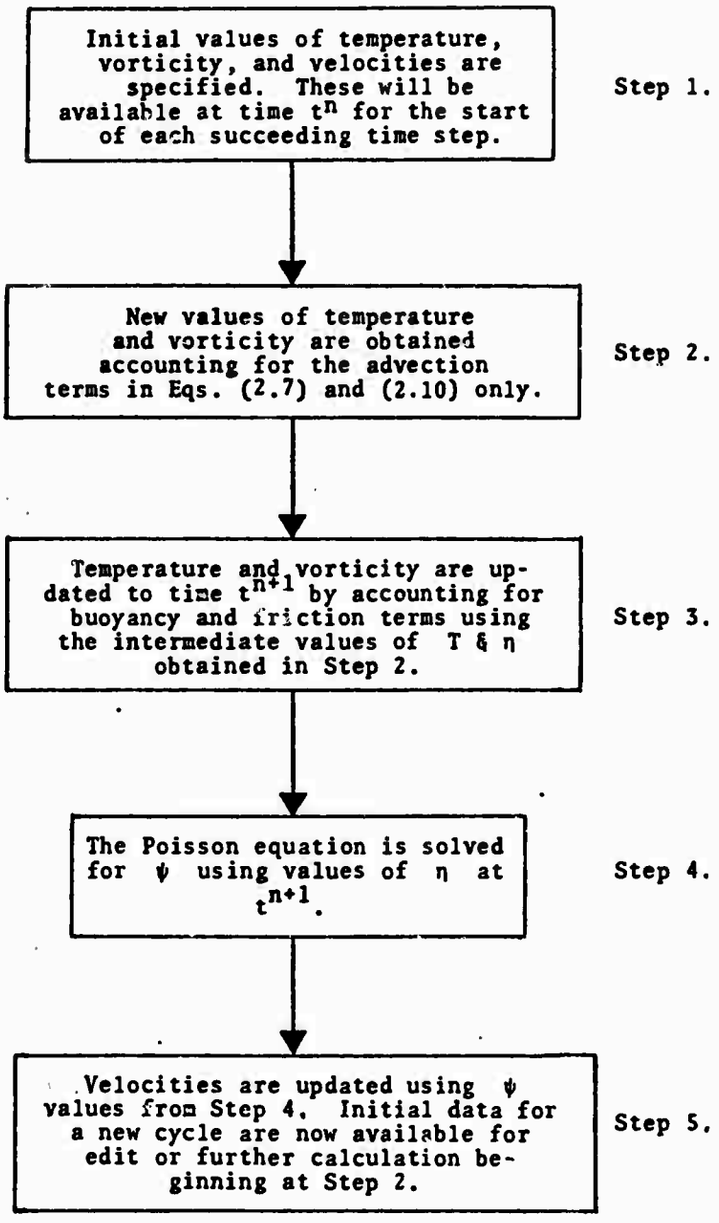


Figure 2.1 – HAIFA Scheme Used in Numerical Integration of Eqs. (2.7), (2.8), and (2.10).

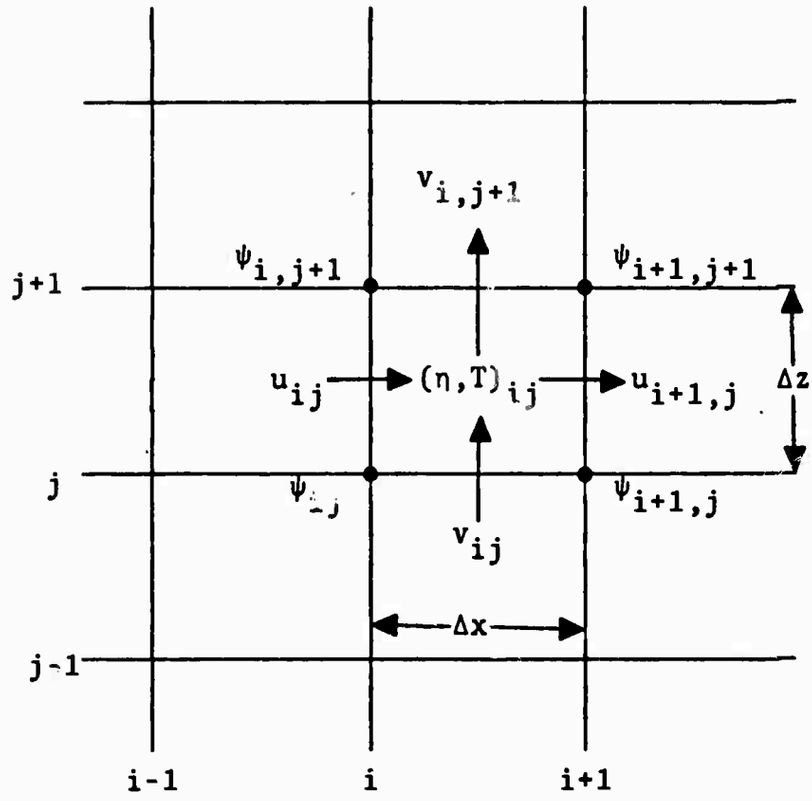


Figure 2.2 – HAIFA Finite Difference Grid.

The stream functions are located at the grid points, the vorticities and temperatures are cell centered and velocities are centered on a grid line located between stream line values. In this way, the velocities defined in finite difference form are:

$$u_{ij} \equiv \frac{\partial \psi}{\partial z} = \frac{\psi_{i,j+1} - \psi_{ij}}{\Delta z} \tag{2.11}$$

$$v_{ij} \equiv - \frac{\partial \psi}{\partial x} = - \frac{\psi_{i+1,j} - \psi_{ij}}{\Delta x} \tag{2.12}$$

2.2.2 The Advection Scheme

The advection of temperature and vorticity in HAIFA is calculated using either the second or fourth order scheme of Crowley.[1] The selection of the second or fourth order scheme is optional and is determined by the trade-off between accuracy and computing time. The schemes chosen are written in conservation form and are based on forward time differences and centered space differences. Test calculations performed by Crowley indicated that for the same order of accuracy, the conservation form produced more accurate solutions than the advection form.

In the conservation form, the time derivative and advection terms of the vorticity or temperature equation may be written as

$$\frac{\partial \phi}{\partial t} + \frac{\partial (u\phi)}{\partial x} + \frac{\partial (v\phi)}{\partial z} = S \quad , \tag{2.13}$$

where ϕ is either T or η and S is the source term.

In two dimensions a splitting technique is used; the calculational scheme calls for solving a one-dimensional equation twice, i.e., the net flux of vorticity or temperature is solved for in the horizontal, the quantity solved for in the zone being updated due to this flux and the procedure is then repeated in the vertical direction using the partially updated values. The equation for the flux across the boundary j written in finite difference form (second order accurate) is

$$F_j = \frac{\Delta x}{\Delta t}\left[\frac{\alpha_j}{2}(\phi_j + \phi_{j-1}) - \frac{\alpha_j^2}{2}(\phi_j - \phi_{j-1})\right] \tag{2.14}$$

where $\alpha_j = u_j \, \Delta t/\Delta x$.

The net change in the variable ϕ in the cell ij due to advection in the horizontal is then

$$\phi^{n+1} - \phi^n = \frac{\Delta t}{\Delta x}(F_j - F_{j+1}) \quad . \tag{2.15}$$

The corresponding fourth order scheme for the flux across the boundary j is

$$F_j = \frac{\Delta x}{\Delta t}\left\{\frac{\alpha_j}{16}\left[9(\phi_j + \phi_{j-1}) - (\phi_{j+1} + \phi_{j-2})\right]\right.$$

$$- \frac{\alpha_j^2}{48}\left[37(\phi_j - \phi_{j-1}) - (\phi_{j+1} - \phi_{j-2})\right]$$

$$- \frac{\alpha_j^3}{12}\left[(\phi_j + \phi_{j-1}) - (\phi_{j+1} - \phi_{j-2})\right]$$

$$\left. + \frac{\alpha_j^4}{24}\left[3(\phi_j - \phi_{j-1}) - (\phi_{j+1} - \phi_{j-2})\right]\right\} \quad . \tag{2.16}$$

The numerical stability of these equations is discussed in Section 2.3 of this report. The accuracy, as discussed by Crowley, is found by expanding the quantities in Taylor series, both in time and in space. The result gives the solution of the variable ϕ at the new time accurate to order Δt^2 in time. The time derivative of the finite difference form of the differential equation is thus accurate to order Δt^3 in time. The second order scheme, Eq. (2.14), has a truncation of order Δx^3 and the fourth order scheme, Eq. (2.16), is accurate to Δx^5 in space.

2.2.3 Update of Other Terms in the Vorticity and Energy Equations

The vorticity equation has two additional terms besides the advective terms. In general, central differences are used in the numerical scheme. The buoyancy term

$$-\frac{g}{T_o}\frac{\partial T'}{\partial x}$$

is expressed as

$$-\frac{g}{T_{o_j}}\frac{T_{i+1,j} - T_{i-1,j}}{2\Delta x} \quad . \tag{2.17}$$

The diffusion term $k_v \nabla^2 \eta$ is expressed as

$$k_v \Big\{(\eta_{i,j+1} - 2\eta_{ij} + \eta_{i,j-1})(\frac{\Delta t}{\Delta y^2})$$

$$+ (\eta_{i+1,j} - 2\eta_{ij} + \eta_{i-1,j})(\frac{\Delta t}{\Delta x^2})\Big\} \quad . \tag{2.18}$$

The energy equation is handled in a similar manner. The diffusion term $k_t \nabla^2 T$ is expressed as in Eq. (2.18) with η replaced by T. The remaining term in the energy equation

$$-w\frac{\partial T_o}{\partial y} + \Gamma$$

is calculated using centered quantities. The term in brackets is calculated analytically from input data and the

velocity is expressed as an averaged quantity

$$w = \frac{w_{i,j+1} + w_{ij}}{2} \quad .$$

2.2.4 Solution of the Poisson Difference Equation by Finite Fourier Transform

The solution of the Poisson equation by means of Fourier transform results in a direct (or exact) solution of the difference equations and their boundary values. In the current version of the subroutine there are some limitations on the generality of the solution; the spatial interval Δx must be constant (see Section 3.3 for variable Δz). The solution must be periodic in the x-direction and prescribed values of the stream function are to be maintained on the top and bottom boundaries of the rectangular region. How the boundary is modified from the rectangular shape is discussed in the following section.

A second order finite difference approximation to the Poisson equation $\nabla^2\psi = \eta$ is obtained by replacing the second derivative operator by a centered second difference operator.

$$\frac{\delta_x^2 \psi_{ij}}{(\Delta x)^2} + \frac{\delta_z^2 \psi_{ij}}{(\Delta z)^2} = \eta_{ij} \qquad \begin{matrix} i = 1,2, \ldots, I \\ j = 2, \ldots, J-1 \end{matrix} \quad , \tag{2.19}$$

where

$$\delta_x^2 \psi_{ij} = \psi_{i+1,j} - 2\psi_{ij} + \psi_{i-1,j}$$

and

$$\delta_z^2 \psi_{ij} = \psi_{i,j+1} - 2\psi_{ij} + \psi_{i,j-1} \quad .$$

Boundary conditions are imposed as follows:

At the bottom of the mesh,

$$\psi_{i,1} = \alpha_i \quad , \qquad i = 1, \ldots, I \ .$$

At the top of the mesh,

$$\psi_{i,J} = \beta_i \quad , \qquad i = 1,2, \ldots, I \ .$$

The cyclic boundary conditions in the horizontal are,

$$\psi_{o,j} = \psi_{I,j} \quad \text{and} \quad \psi_{1,j} = \psi_{I+1,j} \quad , \quad j = 2, \ldots, J-1 \ .$$

We introduce an orthonormal base set of functions having cyclic properties on the index i:

$$w_{ik} = \sqrt{2/I} \cos \frac{2\pi ki}{I} \quad , \qquad I \text{ is even}$$

$$w_{i,I-k} = \sqrt{2/I} \sin \frac{2\pi ki}{I} \quad , \qquad i = 1,2, \ldots, I \ .$$

$$w_{i,I} = 1/\sqrt{I} \quad ,$$

$$k = 1,2, \ldots, \frac{I}{2} - 1 \ .$$

$$w_{i,I/2} = 1/\sqrt{I} \cos i \quad ,$$

These are the finite Fourier functions which have the properties,

$$\sum_{i=1}^{I} w_{ik} \, w_{i\ell} = \delta_{k\ell}$$

and the analogous cyclic boundary conditions are valid in the horizontal. They also have the property that they are eigenfunctions of the central second difference operator

$$\delta_x^2 w_{ik} = -\lambda_k^2 w_{ik}$$

where $\lambda_k = 2 \sin \pi k/I$. These functions are complete functions on the interval $i = 1,2, \ldots, I$. Consequently, an arbitrary function f_i on this space can be represented

$$f_i = \sum_{k=1}^{I} a_k w_{ik}$$

where

$$a_k = \sum_{i=1}^{I} f_i w_{ik} \quad .$$

We are now ready to consider Eq. (2.19) from the point of view of Fourier transformation. The vorticity and stream function are represented as Fourier series as follows:

$$\eta_{ij} = \sum_{k=1}^{I} b_{kj} w_{ik} \quad , \quad \text{where} \quad b_{kj} = \sum_{i=1}^{I} \eta_{ij} w_{ik} \quad ,$$

and (2.20)

$$\psi_{ij} = \sum_{k=1}^{I} a_{kj} w_{ik} \quad , \quad \text{where} \quad a_{kj} = \sum_{i=1}^{I} \psi_{ij} w_{ik} \quad .$$

Substituting into Eq. (2.19) we obtain

$$\sum_{k=1}^{I} w_{ik} \left[\left(-\frac{\lambda_k^2}{(\Delta x)^2} + \frac{\delta_z^2}{(\Delta z)^2} \right) a_{ik} - b_{jk} \right] = 0 \quad .$$

Multiplying by $w_{i\ell}$ and summing over i gives

$$\left(-\frac{\lambda_\ell^2}{(\Delta x)^2} + \frac{\delta_z^2}{(\Delta z)^2} \right) a_{j\ell} = b_{j\ell} \quad , \qquad \begin{matrix} j = 2, \ldots, J-1 \, , \\ i = 1,2, \ldots, I \, . \end{matrix} \tag{2.21}$$

The values of $a_{1,\ell}$ and $a_{J,\ell}$ required by Eq. (2.21) are obtained from the boundary values

$$\begin{aligned} a_{1,\ell} &= \sum_{i=1}^{I} \alpha_i \, w_{i\ell} \quad \text{and} \\ a_{J,\ell} &= \sum_{i=1}^{I} \beta_i \, w_{i\ell} \quad . \end{aligned} \tag{2.22}$$

In Eq. (2.21) the value of the wave number, ℓ, appears only parametrically. For each value of ℓ there is a tri-diagonal equation having fixed values at the end points of the j-interval.

We summarize the procedure for obtaining the direct solution of the Poisson equation, Eq. (2.19), by Fourier transform:

(1) The vorticity and the top and bottom boundary values of the stream function are subjected to Fourier transformation to obtain

$$b_{j\ell} = \sum_{i=1}^{I} \eta_{ij} \, w_{ik} \quad ,$$

$$a_{1,\ell} = \sum_{i=1}^{I} \alpha_i \, w_{i\ell} \quad , \text{ and}$$

$$a_{J,\ell} = \sum_{i=1}^{I} \beta_i \, w_{i\ell}$$

(2) The Fourier components of the stream function are obtained by solving the tridiagonal system of equations, Eq. (2.21), for $a_{j\ell}$.

(3) The stream function itself is obtained by Fourier synthesis

$$\psi_{ij} = \sum_{\ell=1}^{I} a_{j\ell} \, w_{i\ell} \quad .$$

The quantity I must be even. In order to take maximum advantage of the efficiency of the Fast Fourier Transform, the quantity I should also be a power of 2.

2.2.5 The FFT Solution of the Poisson Equation Having Non-Rectangular Boundaries

In order to represent a mountain within the computational grid it is necessary to depart from rectangular boundaries. A modification of the solution algorithm using the FFT is necessary to take account of the specified values of ψ on the mountain contour. The procedure for carrying out this modification of the direct solution of Poisson's equation on an irregular region has been described by Buzbee, et.al.[2]

We consider the case in which there are p internal grid points on which the potential is to be specified. These points constitute the adjacent mesh points lying along the boundary of the mountain which will be assigned the same value of potential (usually zero) as the lower boundary. The first step is to precalculate the stream function contribution at each of the p points of unit vorticity located at each of the points. The solution is then obtained by solving Poisson's equation twice for each cycle. First, Poisson's equation is solved with arbitrary vorticity on the boundary points. The difference between the obtained and desired values of the stream function at each of the p points is used to obtain the corresponding vorticity increments through application of the precalculated matrix. A second solution of Poisson's equation using the incremented vorticity field gives the final value of the stream function within the calculational region.

2.2.6 Description of Poisson Solver Routines

This section describes the subroutines currently used in the HAIFA code to solve the Poisson equation in x-z geometry. The method of solution employs a Fourier transform in the x-direction, solving the resultant set of one-dimensional difference equations (one for each wave number) by Gaussian

elimination in the z-direction and performing the inverse x-direction Fourier transform to obtain the solution. The Cooley-Tukey Fast Fourier Transform (FFT) technique[3] is employed (subroutine COOTUK) with some pre- and post-processing of the data for efficient utilization of the algorithm. In the current version the dependent variable (the stream function ψ in the HAIFA context) is assumed to have cyclic boundary conditions in the x-direction and fixed values at the top and bottom of the grid.

At the beginning of each new calculation, there are references to subroutines which are used only once in each problem. These are called SETUP and OBSET.

SETUP -- This entry references an internal subroutine SET, whose function is to define certain index parameters and required data arrays that are used throughout the calculation by the Poisson solver.

OBSET -- This subroutine is called only when internal boundary conditions are to be applied. Suppose there are p internal points required to have stream function values ψ_1^0, ψ_2^0, ... ψ_p^0. This subroutine computes a $p \times p$ matrix C which has the following property:

> a unit vorticity is placed in the position of internal boundary point j. The value of the independent variable (vorticity) is assumed to be zero at every other point. The Poisson equation solver XYPOIS (see discussion below) is called and returns the influence of that particular unit vorticity on all the other internal boundary points. These influences are put into row j of matrix C.

This procedure is continued until all p internal boundary influences have been computed. Finally, subroutine OBSET forms and stores the inverse matrix C^{-1}.

The controlling subroutine for the Poisson equation solution is named LAPLAC (for the Laplacian symbol ∇^2). This routine is responsible for the solution to both standard boundary condition cases and problems which include internal boundaries.

Each cycle, subroutine LAPLAC averages the cell-centered HAIFA vorticities to provide node-centered vorticities. Then the Poisson equation solver XYPOIS is called to provide the updated values of the stream function. In the case of internal boundaries, one more step is performed in subroutine LAPLAC. Upon the first return from solving the Poisson equation, each internal boundary has a value ψ_i^*, $i=1, \ldots, p$ which in general is not the required value ψ_i^0. A vector $\Delta\psi$ of the differences $\psi_i^0 - \psi_i^*$ is formed. Then, using the inverse matrix C^{-1} formed in subroutine OBSET, one may compute the required modifications Δq_i to the values of the independent variable at each of the p internal boundary points from

$$C^{-1}\begin{pmatrix}\Delta\psi_1\\ \vdots\\ \Delta\psi_p\end{pmatrix} = \begin{pmatrix}\Delta q_1\\ \vdots\\ \Delta q_p\end{pmatrix}$$

The independent variable is so modified, and the XYPOIS package is called once again. The solution returned now contains the correct values for the internal boundary points as well as the other grid points. It remains to discuss the subroutine XYPOIS.

XYPOIS -- This entry is used every calculational cycle to carry out the solution of Poisson's equation. It contains as an argument the values of the inhomogeneous term (here, vorticity) in the interior (nodal) points of the grid, and the fixed values of the dependent variable (here, the stream function) at the top and bottom of the grid. XYPOIS references four internal subroutines:

(1) FFANL (fast Fourier analyzer), which is responsible for carrying out the x-direction transform of vorticity into Fourier components. It processes two rows at a time, so an uncoupling of the row components is required upon return from the FFT routine COOTUK;

(2) GAUSS, which is responsible for solving the resulting z-direction tridiagonal equations for the transform of the dependent variable (see Section 3.3.1);

(3) FFSYN (fast Fourier synthesizer), which is the inverse of FFANL, is responsible for restoring the Fourier components to the new values of the independent variable by another call to subroutine COOTUK. These values, representing the solution to the Poisson equation, are returned to the calling routine (subroutine LAPLAC) in the array containing the original argument list; and

(4) COOTUK, which carries out the Cooley-Tukey fast Fourier transform.

2.3 STABILITY ANALYSIS

A numerical stability analysis of the advection terms in the vorticity and temperature equations has been completed by other researchers. Among them, Crowley[1,4] did a complete analysis for the scheme presently being used in the HAIFA code. The results obtained by Crowley indicate that both his

second and fourth order scheme are stable for all wave numbers if

$$\left|\frac{u\Delta t}{\Delta x}\right| < 1 \quad .$$

Further, the fourth order conservation scheme being used in HAIFA is stable for $(u\Delta t/\Delta x) < 1.5$.

As indicated by Crowley, the schemes both result in amplitude damping and phase lag. For long wavelength disturbances the damping and phase errors are appreciably smaller for the fourth order scheme than for the second order. Comparison tests with a typical mountain wave problem indicated, however, that the differences between fourth and second order solutions are not large. Most of our calculations have been performed with the second order scheme. The criterion built into the HAIFA code is more stringent than any of those noted above, i.e.,

$$\left|\frac{u\Delta t}{\Delta x}\right| < 0.8 \quad .$$

A stability criterion also has been established for the diffusion terms, however, in all problems calculated for this research, the diffusion coefficients are set to zero and thus these terms play no part in the solution.

One unstable region was found using the above criteria in computing the uniform velocity problem discussed in Section 4.3. The details of the instability and the new criteria developed for that problem are also given in that section.

2.4 BOUNDARY CONDITIONS

The initial value problem solved using the HAIFA code requires initial temperature, vorticity and stream function distributions. This is accomplished by prescribing a value of the stream function which is constant in the horizontal direction and which gives the desired horizontal velocity distribution as a function of the vertical coordinate. The vertical velocity component is set to zero. The vorticity at each point in the grid is calculated analytically using the definition

$$\eta = \frac{\partial u}{\partial z} \text{ , since } \frac{\partial v}{\partial x} \text{ is everywhere zero.}$$

The temperature distribution is specified as being horizontally stratified with a lapse rate which may vary with altitude. It is also possible to simulate inversions.

At the beginning of the calculation, with the flow already established, an obstacle is placed in the stream by setting the lower surface streamline to coincide with the mountain surface. A rigid lid (constant streamline) is imposed on the upper boundary of the problem. Figure 2.3 indicates these boundary conditions in graphical form.

The boundary condition imposed at the sides of the grid assumes the flow to be cyclic, i.e., the stream function at each vertical grid line j on the left side of the grid is set equal to the corresponding stream function at the right side of the grid. Mathematically, this can be expressed as $\psi_{1j} = \psi_{n+1,j}$. A graphical explanation of this boundary condition is also given in Figure 2.3.

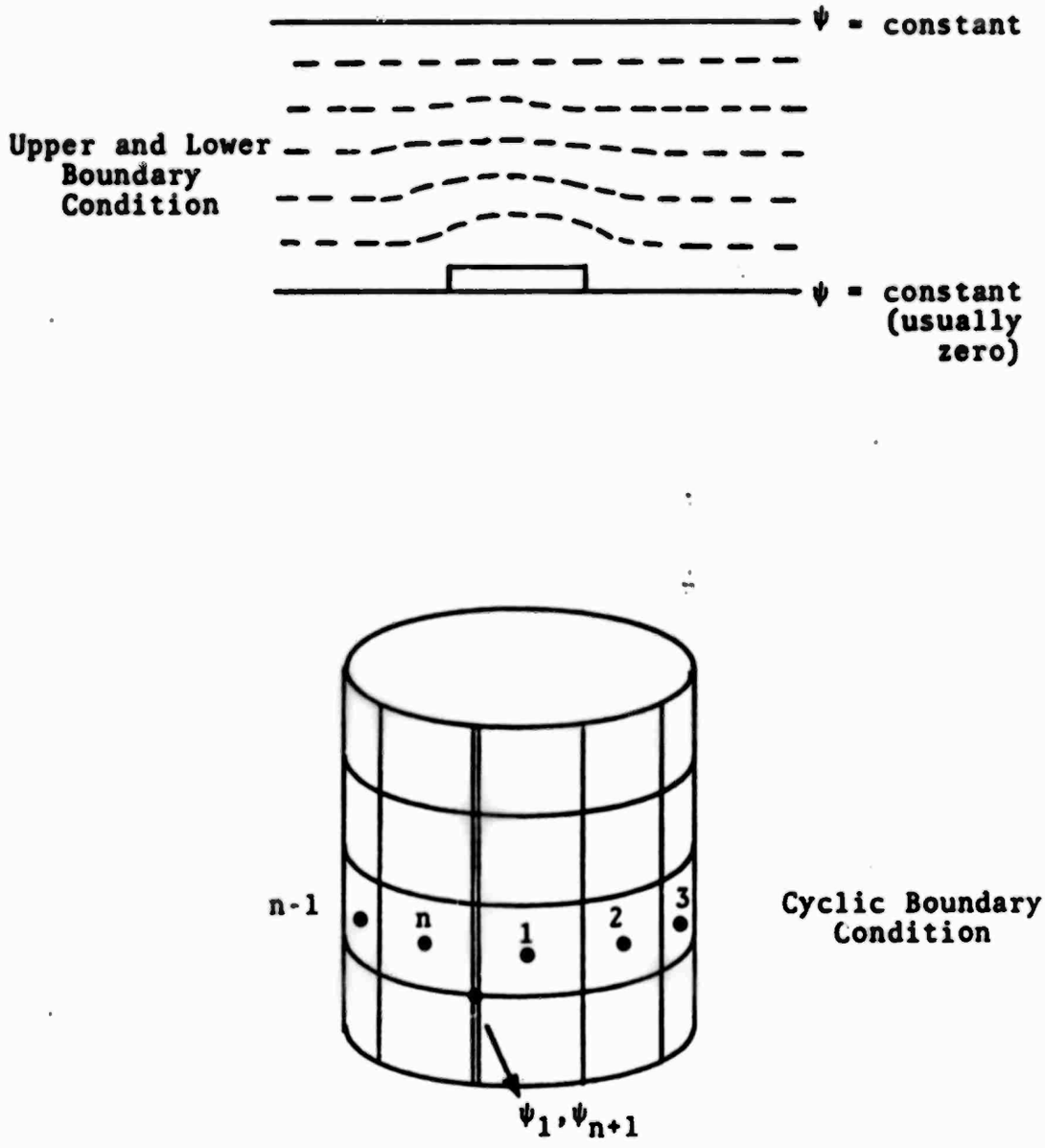


Figure 2.3 – Schematic of HAIFA Boundary Conditions.

One further boundary condition is necessary to obtain the transient solution. The vorticity equation requires that the temperature gradient in the x-direction be specified at the cell center bounded by the obstacle. This requires a value for the temperature perturbation on the obstacle boundary. The assumption is made that the air immediately next to the mountain has risen from the bottom of the grid. The temperature of the air alongside the mountain is thus given by

$$T_m = T_o - \Gamma \cdot z$$

where

T_m = the temperature along the vertical mountain boundaries

T_o = the temperature at ground level

Γ = the dry adiabatic lapse rate

z = the distance above ground level.

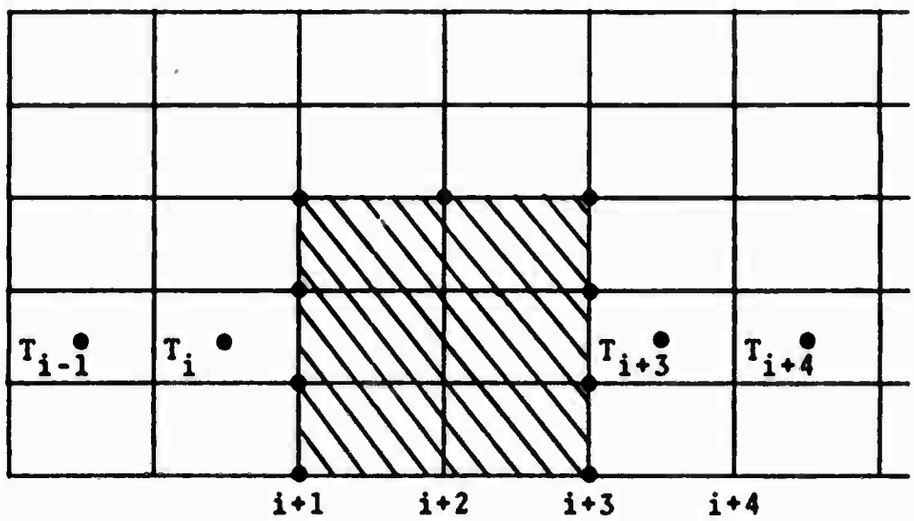
Since the initial temperature profile (T_i) is given as an analytic function of z, the temperature perturbation along the mountain is

$$T'_m = T_o - \Gamma \cdot z - T_i \quad .$$

Referring to the example of a two-cell thick mountain in the figure below, $\partial T'/\partial x$ at the cell centers adjacent to the obstacle are calculated as

$$\frac{\partial T'_i}{\partial x} = \frac{T'_m - \frac{T'_i + T'_{i-1}}{2}}{\Delta x} \quad ,$$

$$\frac{\partial T'_{i+3}}{\partial x} = \frac{\frac{T'_{i+3} + T'_{i+4}}{2} - T_m}{\Delta x} \quad .$$



At cycle zero (time equal to zero), these boundary conditions are then used to determine the new distribution of streamlines within the calculational grid. This completes the required information to start the computation.

2.5 HAIFA CODE DESCRIPTION

A flow chart giving the calculational sequence of the HAIFA code is displayed in Figure 2.4. A description of how problems are generated and the major subroutines within the code is presented below.

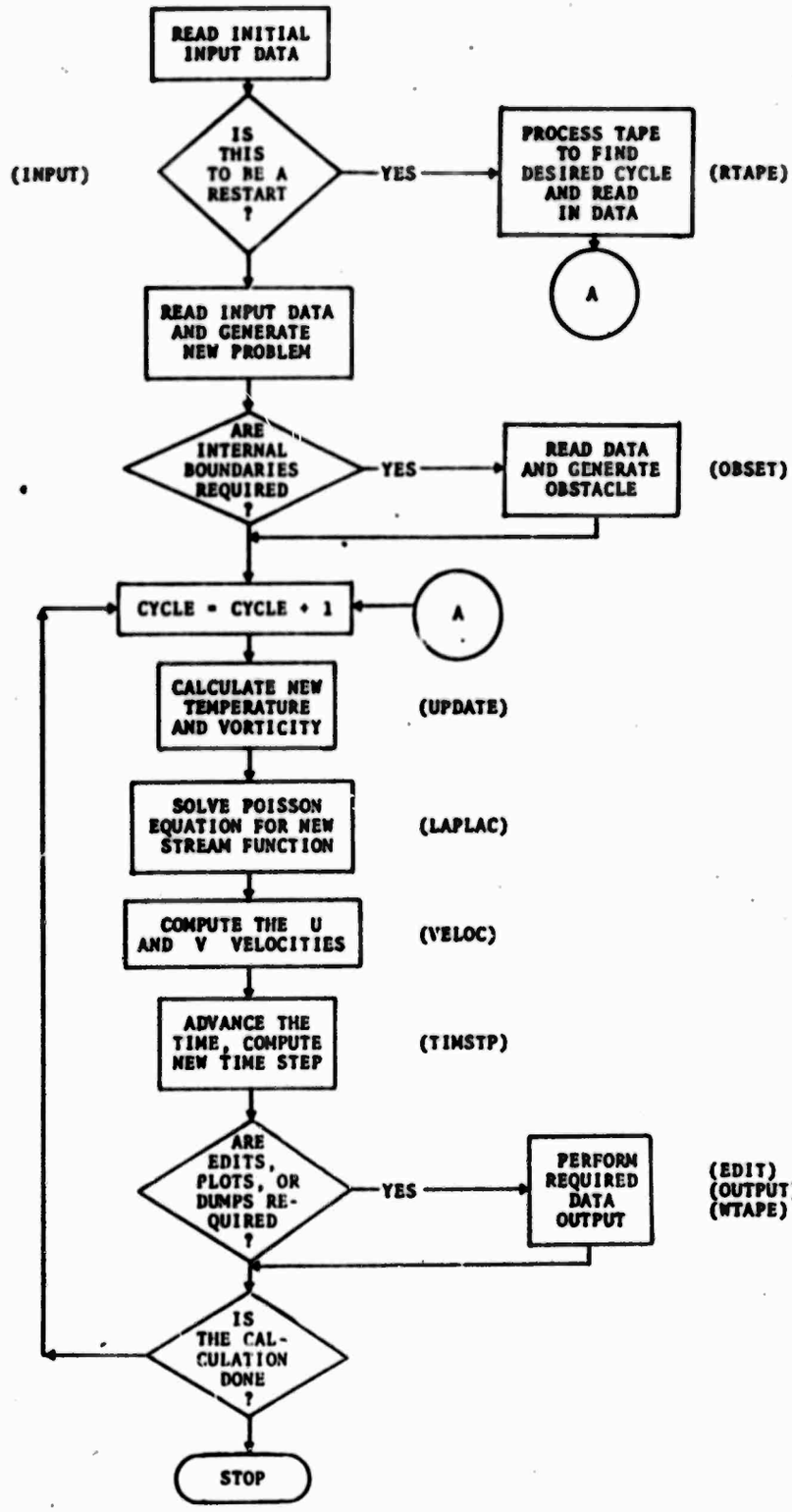


Figure 2.4 – Flow Diagram of HAIFA Code.

2.5.1 Initiating A Calculation

There are two methods for initiating a calculation: generating a new problem, and restarting a partially completed calculation from a data tape. These are controlled by subroutine INPUT.

Generating a New Problem -- Subroutine INPUT reads all input data and sets up several constants which will be used in the calculation. The initial streamline distribution is computed from a series of input parameters, MT , DT1 , DT2 , DT3 , DT4 , and ZETA such that

$$\psi(z) = DT1 + DT2 \cdot z^{MT} + DT3 \cdot z^{(MT+1)} + DT4 \cdot \exp(-ZETA \cdot z) \quad .$$

These parameters define the horizontal velocity distribution

$$u(z) = \frac{\partial \psi}{\partial z} = MT \cdot DT2 \cdot z^{(MT-1)} + (MT+1) \cdot DT3 \cdot z^{MT} - DT4 \cdot ZETA \cdot \exp(-ZETA \cdot z) \quad .$$

The initial vorticities are found from differentiating the above expression with respect to z , i.e., $\eta = \partial u/\partial z$ since $\partial v/\partial x$ is everywhere zero at time equal zero.

The initial temperature distribution is set in a similar fashion using the input parameters KT , AT1 , AT2 , AT3 , AT4 , and ALPHA.

$$T(z) = AT1 + AT2 \cdot z^{KT} + AT3 \cdot z^{(KT+1)} + AT4 \cdot \exp(-ALPHA \cdot z) \quad .$$

Internal Boundaries – The input variable NOBS defines the number of internal points which are to have fixed streamfunction values. A series of data cards specifying the grid points and the associated ψ values are read if NOBS $>$ 0 .

Such internal boundary points are used to define grid obstacles, which are outlined by a series of connected points. Typically, the fixed value of ψ assigned to the obstacle points is the lower boundary streamfunction value. The requested initialization of the streamfunction, vorticities, and velocities in the case of internal boundaries is handled by subroutine OBSET.

Restarting A Calculation – The option to restart a calculation is keyed by the input parameter RESTRT. If it is non-zero in value, the data tape is scanned in subroutine RTAPE until the cycle requested by input parameter ISTART is found. The values of the necessary calculational variables of the requested cycle are then read, and the computation is continued.

2.5.2 Major Subroutines in the Main HAIFA Calculational Loop

UPDATE -- UPDATE is used to solve the conservative equations for vorticity and temperature. Crowley's second order or fourth order scheme is called from this subroutine to calculate the advection terms. This scheme is described in Section 2.2.2 of this report.

LAPLAC -- The Poisson equation relating the stream function and the vorticity is solved using this subroutine as the controlling program. The details of the Poisson solver are given in Section 2.2.4.

VELOC -- The updated stream function values are differenced in z-space to provide the horizontal velocity field u , and in x-space to provide the vertical velocity field v.

PRTTST -- This subroutine defines the type of output required in each cycle, viz, plots, large edits, and/or data dumps on tape are available options with this program.

TIMSTP -- The TIMSTP subroutine calculates a time step to be used in the calculation limited by the numerical stability criterion. The stability criterion is outlined in Section 2.3.

3. MODIFICATIONS TO HAIFA

The HAIFA equations described in the preceeding sections are limited in that the formulation has been simplified both from the mathematical and physical points of view. In Section 3 we discuss several investigations to generalize both the mathematical and physical aspects of the code. The three programs described below are currently being tested and are approaching operational status. Additional features are to be incorporated in the latter part of the contract; they are discussed in Section 6.

3.1 COMPRESSIBILITY

3.1.1 Derivation of the Differential Equations

The use of HAIFA for the investigation of mountain waves is appropriate in that the effects of buoyant stability and dynamics are taken into account, but its applicability is restricted by the incompressibility of the flow. In particular, if the height of the mountain range is comparable with the atmospheric scale height there will be effects induced by the expansion experienced by an air packet in being lifted over the mountain.

The effects of compressibility are to be determined through the use of a new code developed with which problems including this effect may be run and the results compared with HAIFA calculations. Several objectives were sought in arriving at a method of accomplishing this task. They are discussed below.

(1) Sound waves should be excluded from the numerical solutions in order to permit efficient calculations having time intervals comparable with material displacement through a space interval.

(2) Compressibility effects should be retained.

(3) The scheme should be formulated in physical variables to facilitate addition of new physical effects (such as Coriolis force or water vapor).

(4) Conservative difference equations should be sought.

(5) The scheme should retain a mathematical form similar to HAIFA to make programming and check-out as speedy as possible.

The "anelastic" equations of Ogura[5] meet some of the above criteria and will be compared further below. However, the anelastic equations do not allow an arbitrary atmospheric stratification, do not include the change in density due to temperature perturbations and are formulated in problem-dependent variables. These limitations can be avoided, as indicated below.

The equations for inviscid fluid flow on a non-rotating earth (additional terms will be discussed later) are written in conservative form as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} = 0 \quad , \tag{3.1}$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u^2}{\partial x} + \frac{\partial \rho vu}{\partial y} + \frac{\partial \rho wu}{\partial z} + \frac{\partial p}{\partial x} = 0 \quad , \tag{3.2}$$

$$\frac{\partial \rho v}{\partial t} + \frac{\partial \rho uv}{\partial x} + \frac{\partial \rho v^2}{\partial y} + \frac{\partial \rho wv}{\partial z} + \frac{\partial p}{\partial y} = 0 \quad , \tag{3.3}$$

$$\frac{\partial \rho w}{\partial t} + \frac{\partial \rho uw}{\partial x} + \frac{\partial \rho vw}{\partial y} + \frac{\partial \rho w^2}{\partial z} + \frac{\partial p}{\partial z} = -g\rho \quad . \tag{3.4}$$

For a perfect gas, $\frac{1}{\rho} = \frac{RT}{p}$ and the energy equation can be written

$$\frac{dT}{dt} = \frac{1}{\rho C_p} \frac{dp}{dt} \quad . \tag{3.5}$$

For the adiabatic, inviscid non-rotating motion of a perfect gas we have the equations of motion given by Eqs. (3.1) through (3.5).

We now consider the equations in two spatial dimensions, the vertical and down-wind directions, assuming that there is no dependence of any quantity on the y-direction. The equations become

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho w}{\partial z} = 0 \quad , \tag{3.6}$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u^2}{\partial x} + \frac{\partial \rho w u}{\partial z} + \frac{\partial p}{\partial x} = 0 \quad , \tag{3.7}$$

$$\frac{\partial \rho w}{\partial t} + \frac{\partial \rho u w}{\partial x} + \frac{\partial \rho w^2}{\partial z} + \frac{\partial p}{\partial z} = -g\rho \quad , \tag{3.8}$$

$$\frac{dT}{dt} = \frac{1}{\rho C_p} \frac{dp}{dt} \qquad , \qquad p = \rho RT \quad . \tag{3.9}$$

We consider the problem in which the mountain perturbs an initially steady state of the atmosphere, $w_o = 0$, $u_o = u(z,t=0)$, $T_o = T(z,t=0)$, $p_o = p(z,t=0)$, the normal initial conditions previously discussed. These initial values are related to each other through the static atmosphere equation,

$$\frac{\partial p_o}{\partial z} = -g\rho_o = -\frac{gp_o}{RT_o} , \quad \text{or} \quad \frac{\partial \ln p_o}{\partial z} = -\frac{g}{RT_o} . \tag{3.10}$$

The transient solution is obtained by solving the equations of motion starting with the initial values and imposing boundary values on the motion. In order to eliminate sound waves from the transient solution it is sufficient to set $\frac{\partial \rho}{\partial t} = 0$ in Eq. (3.6). The resulting equation,

$$\frac{\partial \rho u}{\partial x} + \frac{\partial \rho w}{\partial z} = 0 , \tag{3.11}$$

is in suitable divergence-free form for the introduction of a solinoidal function. Since Eq. (3.11) is no longer sufficient to determine how the density changes, it is necessary to introduce an additional equation based on an approximation. We assume that the density can be determined at every position from the perfect gas equation of state in which the pressure takes the value associated with the static atmosphere, p_o, through the relation

$$\rho = \frac{p_o}{RT} . \tag{3.12}$$

The temperature equation can be written in terms of the deviation T' of the temperature from its static value ($T = T_o + T'$).

In addition, the temperature equation can be reformulated so that the advection term assumes a conservative form. Expanding the left-hand term of Eq. (3.9)

$$\frac{dT}{dt} = \frac{\partial T'}{\partial t} + u\frac{\partial T'}{\partial x} + w\frac{\partial T'}{\partial z} + w\frac{dT_o}{dz} \quad ,$$

multiplying by ρ and adding Eq. (3.6) multiplied by T', we obtain

$$\frac{\partial \rho T'}{\partial t} + \frac{\partial \rho T'u}{\partial x} + \frac{\partial \rho T'w}{\partial z} + w\frac{dT_o}{dz} = \frac{1}{C_p}\frac{dp}{dt} \quad .$$

Assuming the pressure to have its static value in the right-hand term of Eq. (3.9),

$$\frac{dp}{dt} = \frac{dp_o}{dt} = w\frac{dp_o}{dz} = -wg\rho_o \quad ,$$

the energy equation becomes

$$\frac{\partial \rho T'}{\partial t} + \frac{\partial \rho T'u}{\partial x} + \frac{\partial \rho T'w}{\partial z} = -w\,\rho\frac{\partial T_o}{\partial z} + \rho_o\Gamma \quad , \qquad (3.13)$$

where we have assumed

$$\frac{\rho_o}{\rho} = \frac{T}{T_o} = \frac{T_o + T'}{T_o}$$

from Eq. (3.12).

The "anelastic" equations of Ogura and Phillips also take account of compressibility effects in the atmosphere and it is of interest to compare the above development with them. The anelastic equations are based on several assumptions:

(1) The potential temperature is almost constant; deviations from constancy are small.

(2) The density appearing in Eq. (3.11) is that associated with a neutrally stratified atmosphere.

(3) The potential temperature appearing in the momentum equations is that of the neutral atmosphere, i.e., it is constant. This assumption corresponds to using a neutral atmosphere density in the advection terms of Eqs. (3.7) and (3.8).

The treatments of the buoyancy term of the momentum equations and the energy equations are the same in the two schemes. Consequently, the proposed scheme is more general in two principal respects; the initial stratification of the atmosphere can be arbitrarily specified, and the effect of temperature changes in the atmosphere are reflected in all of the density terms.

The system of compressibility equations, Eqs. (3.7), (3.8), (3.11), and (3.13), have a form similar to the Boussinesq equations, and can be solved in a similar manner. From Eq. (3.11) a stream-function-like quantity ϕ can be introduced:

$$\rho u = \frac{\partial \phi}{\partial z} \quad , \quad \rho w = - \frac{\partial \phi}{\partial x} \quad . \tag{3.14}$$

In terms of a vorticity-like function ζ ,

$$\zeta = \frac{\partial \rho u}{\partial z} - \frac{\partial \rho w}{\partial x} \quad , \tag{3.15}$$

yielding the same Poisson equation as for the Boussinesq approximation,

$$\zeta = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial z^2} \quad . \tag{3.16}$$

The prognostic equation for ζ is obtained by cross differentiating Eqs. (3.7) and (3.8) and subtracting.

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(u\zeta) + \frac{\partial}{\partial z}(w\zeta) + \frac{\partial}{\partial x}\left(\frac{\partial \phi}{\partial x}\frac{\partial u}{\partial x} + \frac{\partial \phi}{\partial z}\frac{\partial u}{\partial z}\right)$$

$$+ \frac{\partial}{\partial z}\left(\frac{\partial \phi}{\partial x}\frac{\partial w}{\partial x} + \frac{\partial \phi}{\partial z}\frac{\partial w}{\partial z}\right) = g\frac{\partial \rho}{\partial x} = -\frac{g\rho}{T_o + T'}\frac{\partial T'}{\partial x} \quad . \tag{3.17}$$

Eq. (3.17) replaces the vorticity equation of the Boussinesq equations, differing principally in having the additional terms containing the derivations of ϕ, u and w.

3.1.2 Method of Numerical Solution

The system of compressibility equations is seen to be very similar to the HAIFA equations, and in fact only nominal modifications to the HAIFA code were required to produce a compressible low-speed flow code.

Generating Initial Conditions — As with HAIFA, the values of $u_o(z)$ and $T_o(z)$ are specified by input to the code. In addition, the initial surface pressure $P_o(z=0)$ must be specified. The remaining initial pressures are found using the relation of Eq. (3.10),

$$P_o(z) = P_o(z=0)\ \exp -\frac{g}{R}\int_o^z \frac{dz}{T_o(z)} \quad .$$

The initial density profile then follows from

$$\rho_o(z) = \frac{P_o(z)}{RT_o(z)} \quad .$$

The stream-function-like quantity ϕ is formed by integrating

$$\frac{\partial \phi}{\partial z} = \rho_o(z)\ u_o(z) \quad ,$$

and the vorticity-like quantity ζ is initialized from the non-zero term of Eq. (3.15),

$$\zeta = \frac{\partial \rho u}{\partial z}$$

(as in HAIFA, there is no x-dependence of any quantity initially).

Modification of Advection Scheme – The quantities to be advected in the system of compressibility equations are ζ, Eq. (3.17), and $(\rho T')$, Eq. (3.13). Since the equation of continuity is in the form

$$\frac{\partial \rho u}{\partial x} + \frac{\partial \rho w}{\partial z} = 0 \quad ,$$

Crowley's second order scheme for advection was modified to use (ρu) and (ρw) as pseudo-velocities.

This code is now complete and problems will be run in the next six months in order to compare with HAIFA results. In addition, a problem will be calculated using a completely compressible code.

3.2 MOISTURE

3.2.1 Derivation of Equations

In this section the effects of moisture on the equations for Boussinesq fluid flow are discussed. Frequently, atmospheric water in the form of water vapor, cloud water, and precipitation will have important effects on the characteristics of gravity waves caused by mountains. Lee waves are frequently accompanied by clouds which can be expected to modify the stability of the air through the presence of the latent heat of condensation which the cloud water adds to the air. In addition, there are effects discussed by Orville[6,7] of up-slope winds due to high-level heating and evaporation, but these are not of primary interest in our investigation. Consequently, the terms resulting in changes of stability of the air in which clouds are forming are of primary interest.

Radiative heating and cooling of the air has not been taken into account in this discussion, even though the boundary condition on moisture is affected by it. Boundary layer effects at present are largely omitted from HAIFA but are considered in Appendix E. It will be beneficial to incorporate them, together with radiative terms, at a later stage of the code development work.

The HAIFA equations are to be modified to include the effects of moisture by incorporating the following major changes:

(1) the momentum equation is modified by including the effects of moisture in the buoyancy term,

(2) the equation of state for air is changed to include moisture,

(3) the energy equation is modified to include energy changes equivalent to the latent heat of water being given to or taken from the air,

(4) a new equation is added to account for the conservation of moisture in the air excluding rain water, and

(5) a conservation equation is included which expresses the rainwater content in the atmosphere including sources and sinks at the boundaries.

An outline of the derivation of the equations is given below.

The momentum equations and the equation of state for a system with moisture are

$$\frac{du}{dt} = - \frac{1}{\rho_o} \frac{\partial p}{\partial x} \quad , \tag{3.18}$$

$$\frac{dw}{dt} = - \frac{1}{\rho_o} \frac{\partial p}{\partial z} + \frac{\rho}{\rho_o} g(1 + \ell_c + \ell_r) \quad , \tag{3.19}$$

$$p = \rho RT(1 + E_r) \quad . \tag{3.20}$$

These equations have been derived by Orville, and Ogura and Phillips among others. The energy equation is now derived from the first law of thermodynamics in order to redefine some terms used previously, i.e.,

$$Tds = dh - \frac{dp}{\rho} \quad . \tag{3.21}$$

For an adiabatic system, this can be written as

$$\frac{dh}{dt} = \frac{1}{\rho}\frac{dp}{dt} = \frac{1}{\rho}\left(\frac{\partial p}{\partial t} + u\frac{\partial p}{\partial x} + w\frac{\partial p}{\partial z}\right) . \qquad (3.22)$$

In the Boussinesq approximation, the first two pressure terms in brackets are zero and the third term is equivalent to $-wg\rho$ (assuming the hydrostatic approximation),

$$\frac{dh}{dt} = -wg . \qquad (3.23)$$

The enthalpy changes will include energy changes due to both advection of the temperature and latent heat being released or absorbed as the moisture in the air changes phase. The energy equation can thus be expressed as

$$\frac{\partial \tilde{T}}{\partial t} + u\frac{\partial \tilde{T}}{\partial x} + w\frac{\partial \tilde{T}}{\partial z} = -\frac{wg}{C_p} = -w\Gamma$$

where

$$\tilde{T} = T + \frac{Lr}{C_p} .$$

Expressing the temperature as $T = T_o + T'$ where T_o is a mean value which is constant and T' represents all variations of the temperature from this space averaged quantity, i.e.,

$$\tilde{T} = T_o + T' + \frac{Lr}{C_p} = T_o + T'' . \qquad (3.24)$$

Eq. (3.23) becomes

$$\frac{\partial T''}{\partial t} + u\frac{\partial T''}{\partial x} + w\frac{\partial T''}{\partial z} = -w\Gamma \quad . \tag{3.25}$$

A diffusion term may be included in the above equation as $+k_t \nabla^2 T''$ where k_t is a constant coefficient.

The vorticity equation, derived from the momentum and continuity equations, is

$$\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + w\frac{\partial \eta}{\partial z} = g(1 + \ell_c + \ell_r)\frac{\partial}{\partial x}(\rho/\rho_o)$$

$$+ \frac{\rho}{\rho_o} g\frac{\partial \ell_c}{\partial x} + \frac{\rho}{\rho_o} g\frac{\partial \ell_r}{\partial x} \quad . \tag{3.26}$$

Making the Boussinesq approximation and the further restriction that $T_o/(T_o + T') \cong 1$, the equation is

$$\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + w\frac{\partial \eta}{\partial z} = -\frac{g}{T_o}(1 + \ell_c + \ell_r)\frac{\partial T''}{\partial x}$$

$$+ \frac{g}{T_o}\frac{L}{C_p}(1 + \ell_c + \ell_r)\frac{\partial r}{\partial x} + g\frac{\partial \ell_c}{\partial x} + \frac{\partial \ell_r}{\partial x} \quad . \tag{3.27}$$

Eqs. (3.25) and (3.27) and the Poisson equation relating the stream function and the vorticity replace Eqs. (2.7) and (2.10) in the basic HAIFA scheme.

The equation of water vapor conservation is obtained by deriving equations of total water conservation and rain water conservation and taking the difference between them. The total water conservation equation may be obtained by

equating the sum of the time derivative of the total water plus the diffusion of the water carried as cloud water and moisture to zero. Mathematically, this is expressed as

$$\frac{\partial}{\partial t}(\rho Q) = -\nabla\cdot(\rho r\vec{V}) - \nabla\cdot(\rho \ell_c \vec{V})$$

$$- \nabla\cdot \rho\int(\vec{V}-V_D)\ell_r^D \, dD$$

$$+ k\nabla^2 \rho(r+\ell_c) \quad . \tag{3.28}$$

The integral term on the right side of the equation represents rain water advection and fallout as a function of droplet diameter. Several authors including Orville, Kessler,[8] Srivastava,[9] and Armason, et.al.[10] have derived expressions for water droplet formation and precipitation in the atmosphere. At this stage in the development of HAIFA with moisture, we have elected to program the conservation equations as derived by Orville with one or two exceptions noted below and will modify these equations as we derive or discover better prescriptions for each of the terms.

The final equation for total water conservation may be expressed as

$$\frac{\partial Q}{\partial t} = -V\nabla\cdot Q + \ell_r \frac{\partial V_t}{\partial z} + V_t \frac{\partial \ell_r}{\partial z} + \frac{k}{\rho}\nabla^2 \rho(\ell_c+r)$$

$$+ \ell_r V_t + \frac{1}{\rho}\frac{\partial \rho}{\partial z} + \ell_r V_t \frac{1}{\rho}\frac{\partial \rho}{\partial x} + V_t \frac{\partial \ell_r}{\partial x} \quad . \tag{3.29}$$

The last two terms on the right side of Eq. (3.29) are ignored by Orville. At the present stage of our analysis, the magnitude of these terms with respect to others in the equation are is unknown; further study will be made to justify retaining them.

The final equation required to complete our analysis expresses conservation of rain water in the atmosphere. The change of the rain water with respect to time is equal to the advection and fallout of the droplets plus a source term which expresses the conversion of cloud water into rainwater, the growth of rain drops through coalescence, and the evaporation of rain falling through unsaturated air. The production terms also have been derived by the authors already noted. The most satisfying expression seems to be that derived by Orville or Arnason. For consistency, our original equations for the production term will be equivalent to those arrived at by Orville. Modifications will be made where we obtain an improved description of the processes being undergone by the water. The equation may be expressed as

$$\frac{\partial \ell_r}{\partial t} - \ell_r \frac{\partial V_t}{\partial x} - \frac{\ell_r V_t}{\rho} \frac{\partial \rho}{\partial x} - \ell_r \frac{\partial V_t}{\partial z} - \frac{\ell_r V_t}{\rho} \frac{\partial \rho}{\partial z}$$

$$- u \frac{\partial \ell_r}{\partial x} + v \frac{\partial \ell_r}{\partial z} - V_t \frac{\partial \ell_r}{\partial z} - V_t \frac{\partial \ell_r}{\partial x} = P_r \quad . \qquad (3.30)$$

This equation, Eq. (3.30), includes variations of ρ, ℓ_r, and V_t with respect to the horizontal direction. Orville has ignored these, but for completeness and until we can substantiate that they are negligible, they will be carried in our studies. The production term, P_r, is described in detail in Appendix B.

The equation for conservation of water vapor is found by subtracting Eq. (3.30) from Eq. (3.29). The result is

$$\frac{\partial q}{\partial t} + \vec{V}\cdot\nabla q = k_q \nabla^2 q - \ell_r \frac{\partial V_t}{\partial x} - P_r \quad . \tag{3.31}$$

Eqs. (2.8), (3.25), (3.27), (3.29), and (3.31) constitute the complete set of equations to be solved in HAIFA with moisture.

3.2.2 Difference Equations

The difference equations used in HAIFA with moisture are formed in an identical manner as those in the basic HAIFA. All moisture terms are cell centered quantities. The time differences are taken in the forward direction, the advection terms are treated by Crowley's schemes and all other terms are centered in space through appropriate averaging. Since this version of HAIFA is not thoroughly checked out at the time of this report, the finite difference equations will not be presented here. A complete listing of these equations will be a part of the annual report under this contract.

3.3 VARIABLE ZONING IN VERTICAL DIRECTION

The modifications to the basic HAIFA code that will enable it to operate with a mesh of variable spacing in the vertical direction are examined in this section. This modification affords the ability to resolve more finely certain areas without excessively slowing the computation by requiring fine zoning throughout the grid. Modifications to two routines of the code are necessary. They are the Poisson equation solver, and the vertical advection subroutine. Each modification is discussed below.

3.3.1 The Poisson Solver

The use of the Fast Fourier Transform in the horizontal x-direction imposes the limitation that the spatial interval, Δx , be constant. In the vertical direction, however, the solution of the Poisson equation is obtained by Gaussian elimination and is not limited to a constant spatial interval.

The Gaussian elimination subroutine of POISPK solves a system of difference equations approximating

$$\frac{\partial^2 \psi}{\partial z^2} - \alpha \psi = Q \quad . \tag{3.32}$$

The solution of these equations is briefly outlined below:

The finite difference form of Eq. (3.32) may be written as a tridiagonal system

$$A_i \psi_{i+1} + B_i \psi_i + C_i \psi_{i-1} = D_i \quad . \tag{3.33}$$

Letting

$$\psi_i = E_i \psi_{i+1} + G_i \tag{3.34}$$

which implies

$$\psi_{i-1} = E_{i-1} \psi_i + G_{i-1} \quad , \tag{3.35}$$

and substituting into the tridiagonal system, the coefficients E_i and G_i may be expressed as

$$E_i = - \frac{A_i}{B_i + C_i E_{i-1}} \quad , \tag{3.36}$$

$$G_i = \frac{D_i - C_i G_{i-1}}{B_i + C_i E_{i-1}} \quad . \tag{3.37}$$

The finite difference form of Eq. (3.32) for constant vertical zoning is

$$\frac{\psi_{i+1} - [2 + \alpha(\Delta z)^2]\psi_i + \psi_{i-1}}{(\Delta z)^2} = Q_i \quad . \tag{3.38}$$

and the coefficients A_i , B_i , C_i , and D_i are thus equivalent to

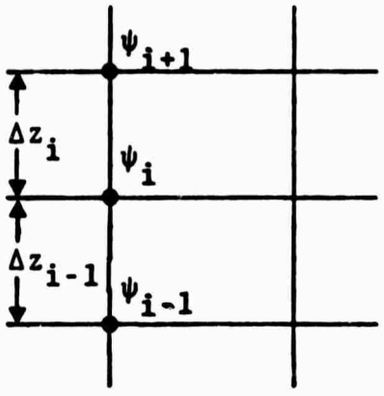
$$\begin{aligned} A_i &= 1/(\Delta z)^2 \quad , \\ B_i &= -\alpha - 2/(\Delta z)^2 \quad , \\ C_i &= 1/(\Delta z)^2 \quad , \\ D_i &= Q_i \quad . \end{aligned} \tag{3.39}$$

Using these coefficients, E_i and G_i can be calculated and thus the ψ_i may be solved for recursively.

With variable zoning the finite difference form of Eq. (3.32) becomes

$$\frac{\dfrac{\psi_{i+1} - \psi_i}{\Delta z_i} - \dfrac{\psi_i - \psi_{i-1}}{\Delta z_{i-1}}}{\dfrac{\Delta z_i + \Delta z_{i-1}}{2}} - \alpha\psi_i = Q_i \quad , \tag{3.40}$$

where the location of ψ and Δz are shown below,



The coefficients A_i , B_i , C_i , and D_i are now equivalent to

$$\begin{aligned} A_i &= 1/(\ z_i(\Delta z_i + \Delta z_{i-1})/2) \\ B_i &= -2/(\Delta z_i\ \Delta z_{i-1}) - \alpha \\ C_i &= 1/(\Delta z_{i-1}\ (\Delta z_i + \Delta z_{i-1})/2) \\ D_i &= Q_i \quad . \end{aligned} \tag{3.41}$$

The values of E_i and G_i are computed using the above coefficients and ψ_i is computed in the same manner as indicated above by Eq. (3.35).

3.3.2 Vertical Advection

The advection schemes discussed previously are valid for uniform zones only. The equivalent scheme for variable size zones is derived below for the Crowley second order scheme. It has been incorporated into a version of HAIFA which is currently being tested. The fourth order scheme will be considered at a later date.

The one-dimensional advection equation in conservation form may be written for flow in the z-direction as follows:

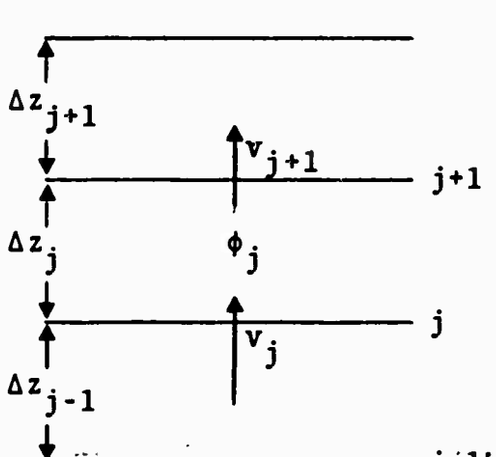
$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial z}(v\phi) = 0 \quad , \tag{3.42}$$

where ϕ is a variable representing the quantity to be advected. Only the one-dimensional equation need be considered due to the splitting technique used in HAIFA.

In finite difference form, Eq. (3.42) is

$$\phi_j^{n+1} = \phi_j^n - \frac{\Delta t}{\Delta z_j}\left[(v\phi)_{j+1} - (v\phi)_j\right] \cdot \tag{3.43}$$

The term $\Delta(v\phi)_j$ requires the flux across the boundary of the j cell (see figure below).



The flux at the $j+1$ boundary may be expressed as $v_{j+1}\,\phi_b$ where ϕ_b represents the value of the variable ϕ at that boundary. Assuming ϕ to vary linearly between zone centers, this flux may be expressed as

$$v\phi)_{\text{boundary}} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} v\phi\, dt = \frac{1}{\Delta t}\int_{z-vt}^{z} \phi\, dz \quad . \tag{3.44}$$

Assuming $\phi = a + bz$, and integrating,

$$v\phi)_{\text{boundary}} = F_{j+1} = v_{j+1}\left[\phi_j + \frac{(\phi_{j+1} - \phi_i)\,\Delta z_j}{\Delta z_{j+1} + \Delta z_j}\right]$$

$$- v_{j+1}{}^2 \Delta t \,\frac{\phi_{j+1} - \phi_j}{\Delta z_{j+1} + \Delta z_j} \quad . \tag{3.45}$$

The new value of ϕ_j^{n+1} can then be expressed using Eq. (3.45) as

$$\phi_j^{n+1} = \phi_j^n + \frac{\Delta t}{\Delta z_j}\left(F_j - F_{j+1}\right) \quad . \tag{3.46}$$

4. TEST PROBLEMS

Several problems have been calculated using the basic HAIFA code. The results of each are presented in this section and comparisons with other results are made where possible. An edit routine to determine the momentum flux (wave drag associated with gravity waves) was written and is described in detail in Appendix C.

Table I summarizes the initial conditions used for each problem. The boundary conditions in each case were those described in Section 2.4 of this report. The grid size consisted of 35 vertical cells by 64 horizontal cells.

4.1 SINGLE WAVE

The atmospheric and horizontal velocity conditions to produce a single gravity wave were arrived at using the results presented on two-dimensional mountain lee waves by Palm and Foldvik.[11] They had established that if the quantity

$$\frac{S}{u^2} - \frac{1}{u}\frac{\partial^2 u}{\partial z^2} \quad ,$$

where S is the stability of the atmosphere, has a value at the ground level which is at least 2.5 times as large as the minimum value (usually located 7-10 km above the ground),

TABLE I

TEST PROBLEM	LAPSE RATE or TEMPERATURE DISTRIBUTION	OBSTACLE/GRID SIZE	HORIZONTAL VELOCITY PROFILE
Single Wave	$\gamma = \frac{1}{2}\Gamma$*	625 meters high × 4.5 kilometers long Δx = 312.5 meters Δy = 1.5 km	Exponential Velocity Profile shown in Figure 4.1.
Two Wave	$\gamma = \frac{1}{2}\Gamma$	625 meters high × 4.5 kilometers long Δx = 312.5 meters Δy = 1.5 km	Exponential Velocity Profil shown in Figure 4.2.
Uniform Velocity	$\gamma = \frac{1}{2}\Gamma$	1.0 kilometer high × 3 kilometers long Δx = 1 km Δy = 1 km	Velocity = 10m/sec
Inversion Layer I	$\gamma = \Gamma$ to 4.5 km ΔT = +4°C @ 4.5 km $\gamma = \frac{1}{2}\Gamma$ above 4.5 km	1.5 kilometers high × 4.5 kilometers long Δx = 1.5 km Δy = 1.5 km	Exponential Velocity Profile to 13.5 km, u at (z = 750m) = 43 m/sec, u at (z = 14.25 km) = 146 m/sec. u = constant above 14.25 km (set equal to velocity at 14.25 km resulting from exponential profile).
Tropopause	$\gamma = 0.65\Gamma$ to 8.25 km $\gamma = 0$ above 8.25 km	1.5 kilometers high × 4.5 kilometers long Δx = 1.5 km Δy = 1.5 km	Exponential Velocity Profile to 8.25 km, u at (z = 750m) = 26 m/sec, u at (z = 9.75 km) = 60 m/sec. u = constant above 9.75 km (set equal to velocity at 9.75 km resulting from exponential profile).

* Γ = adiabatic lapse rate -- T_o for all problems assumed to be 300 °K.

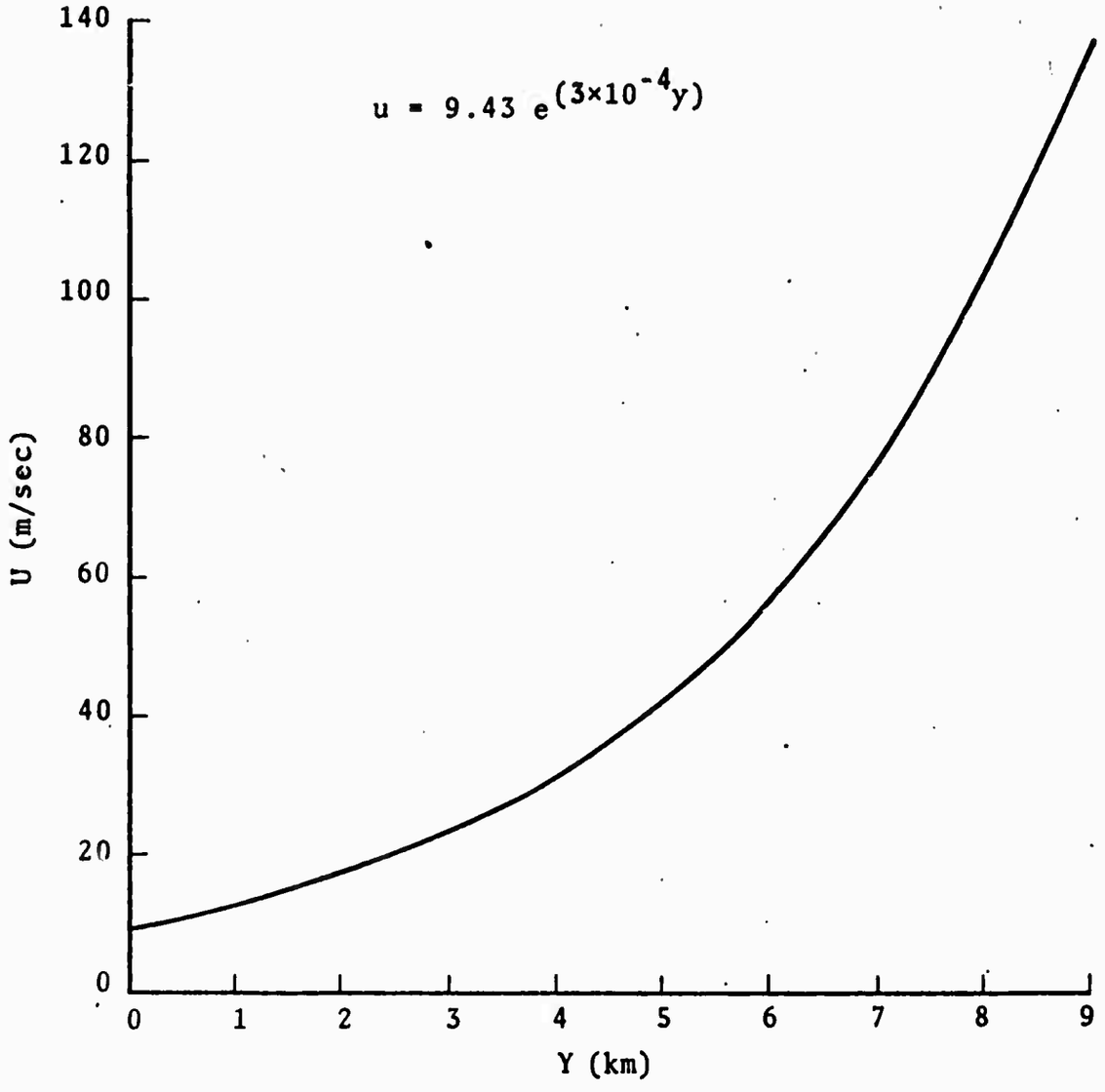


Figure 4.1 – Single Wave Velocity Profile.

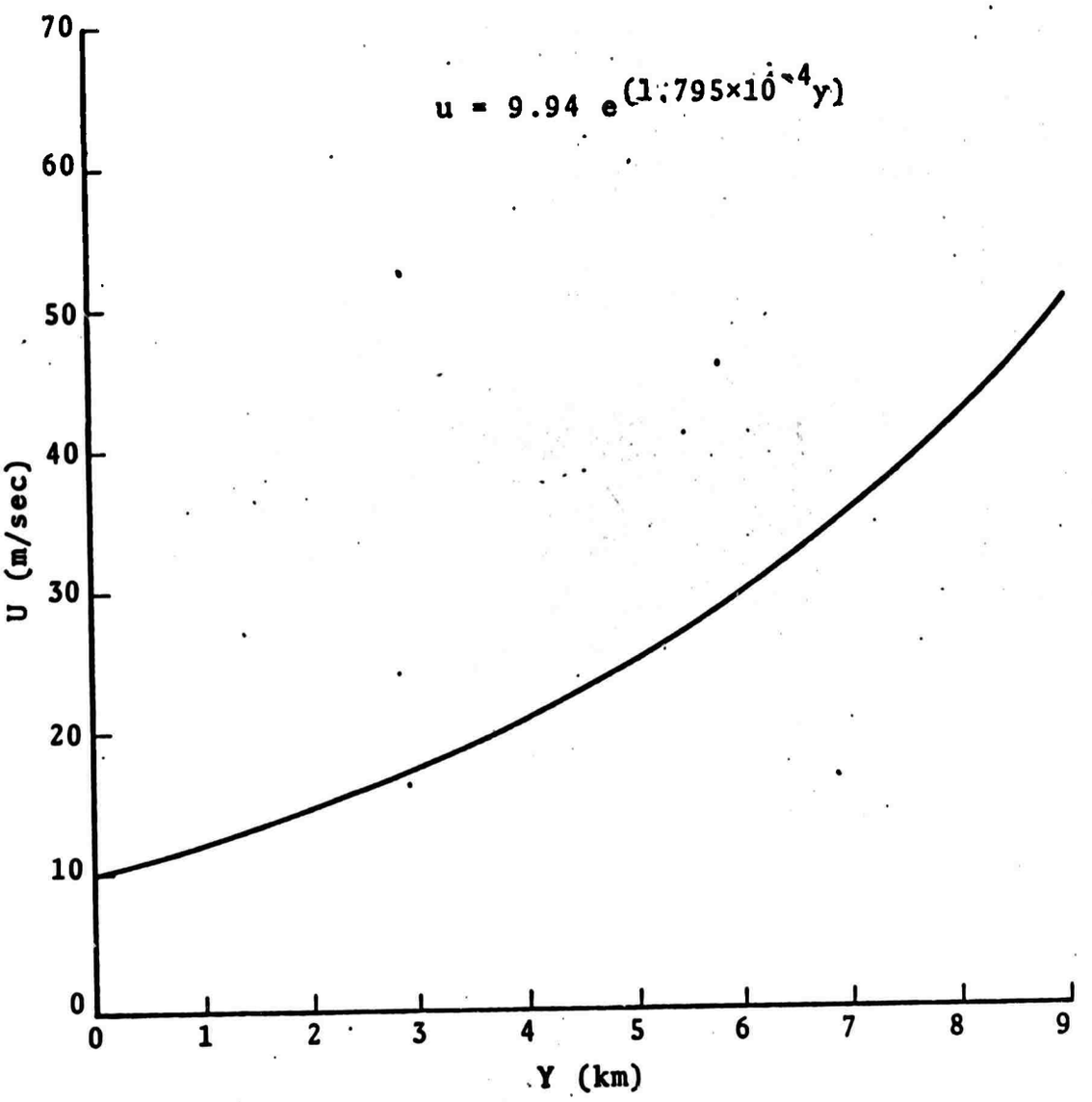


Figure 4.2 – Two Wave Velocity Profile.

the wave motion in the lower troposphere depends only on the wind profile and the stability. This condition is almost always satisfied when mountain waves occur. A diagram giving the expected wave lengths of lee waves under various stability and wind profiles was presented. In particular, regions of one and two waves were indicated. Using this diagram, a single wave of approximately 16 km in length was predicted for a lapse rate equal to one-half the dry adiabatic value (see Figure 4.3), and the exponential velocity profile shown in Figure 4.1.

The numerical results calculated using HAIFA are shown in Figures 4.4 through 4.7 as streamlines and vertical velocity contours at several times up to 1-1/4 hours. The measured wave length from Figure 4.5 or 4.7 is approximately 15 km. As can be observed from the results, only one wave did form during the time the problem was run. The cyclic boundary condition prevented any further computation due to disturbances created by the obstacle in the flow stream being introduced into the main flow upstream of the mountain. Some interference with the upper boundary positioned at 10.9 km may also be seen at the latest times.

The momentum edits $\overline{u'v'}$ (see Appendix C) located one cell or 312.5 meters above the mountain top are shown in Figures 4.8 and 4.9 for various lengths used in obtaining the horizontal averages. The qualitative result obtained from these figures indicates a decrease in the edited quantity as the length used in the averaging length is increased, i.e., a lower amount of drag is created by the mountain. One exception appears, however; this can be noted as a cross over

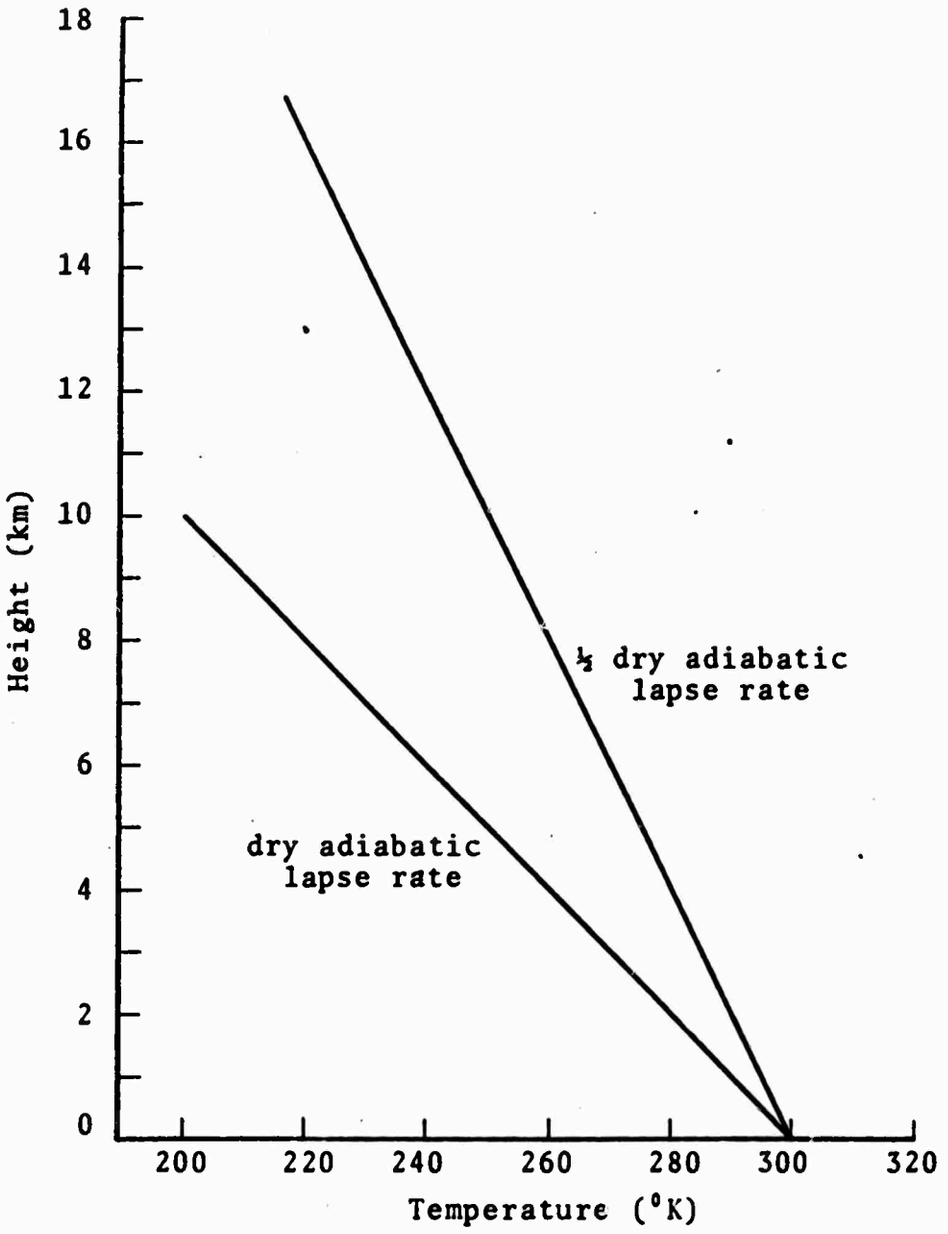


Figure 4.3 – Initial Temperature Profiles used in Test Problems.

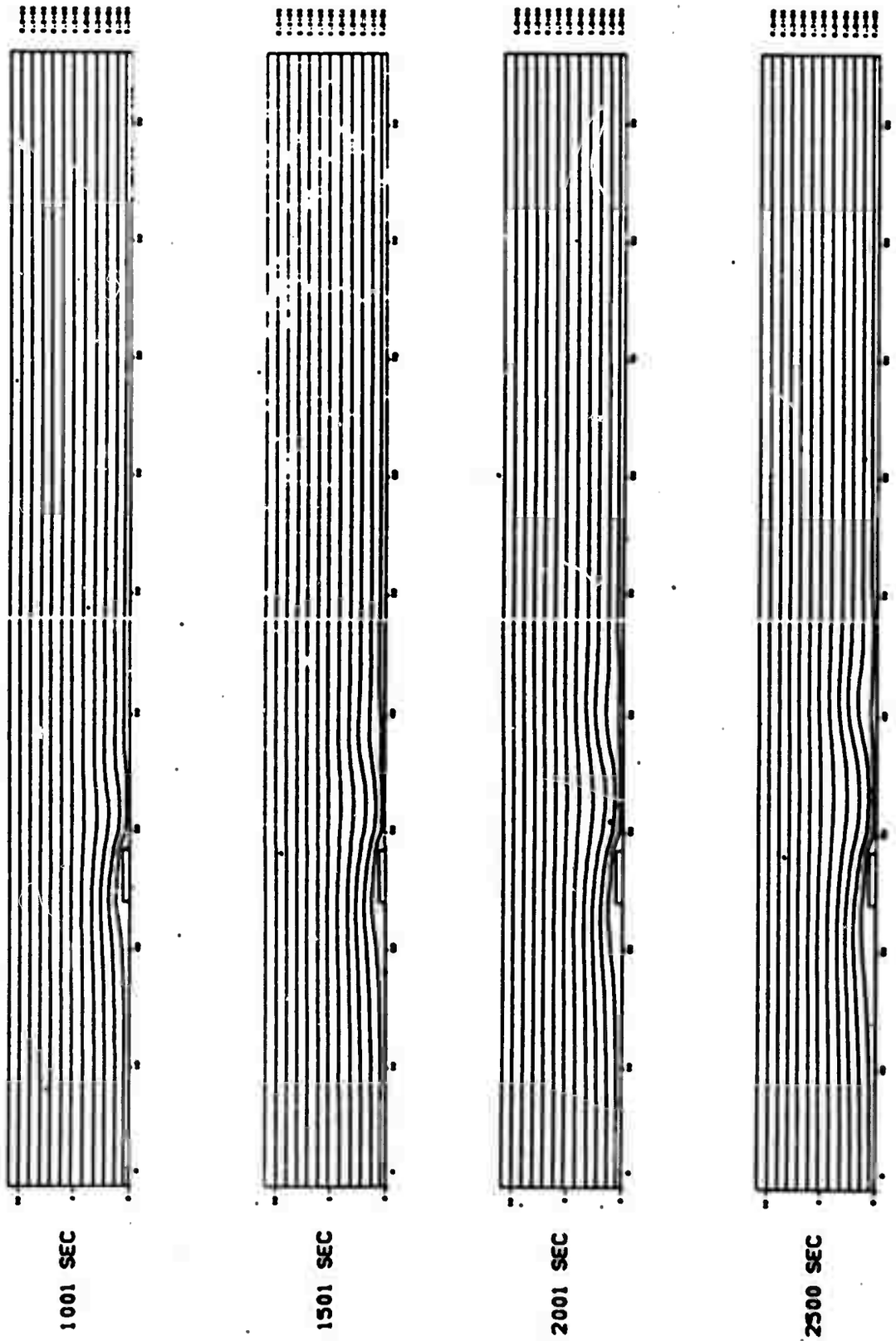


Figure 4.4 – Streamlines from Single Wave Problem.

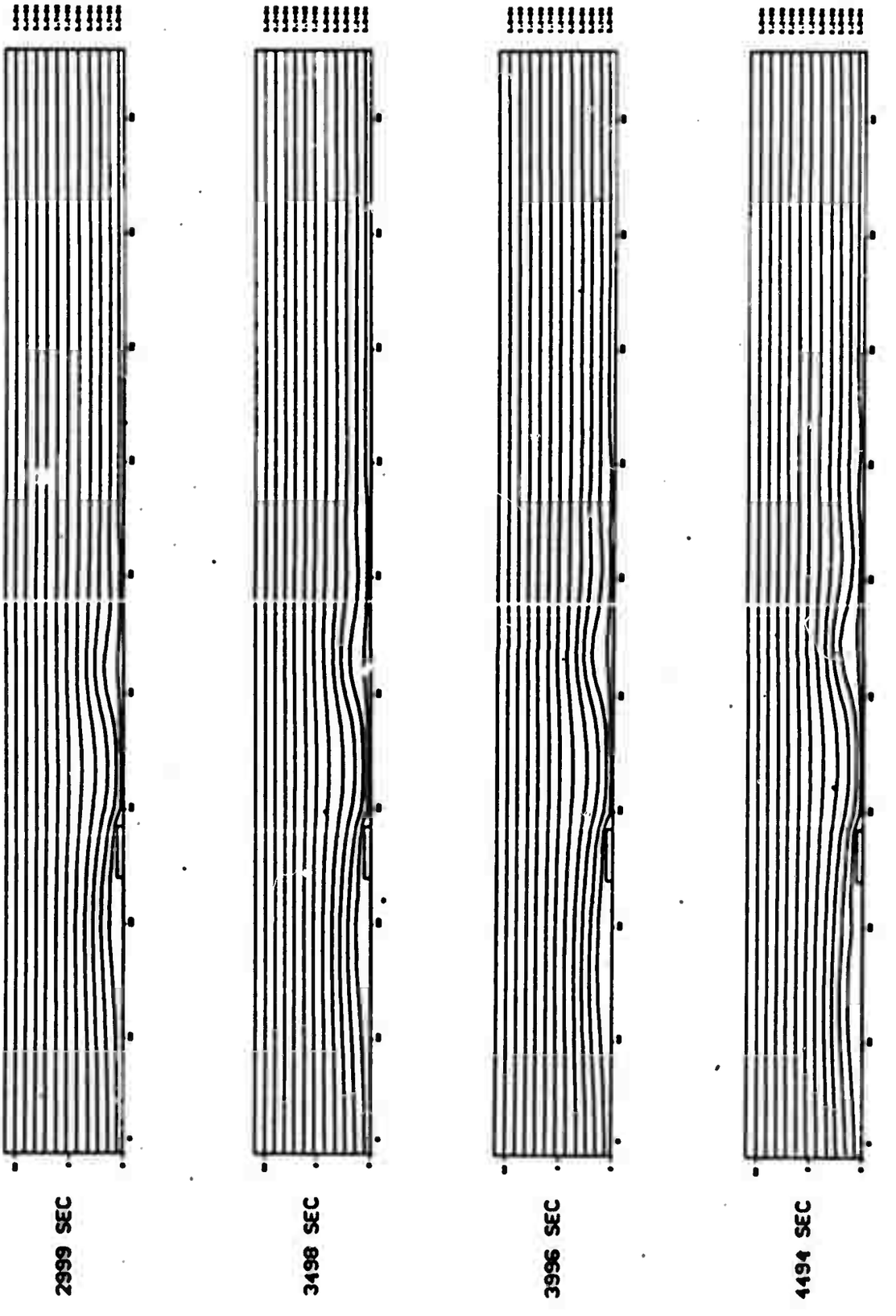


Figure 4.5 – Streamlines from Single Wave Problem (page 2).

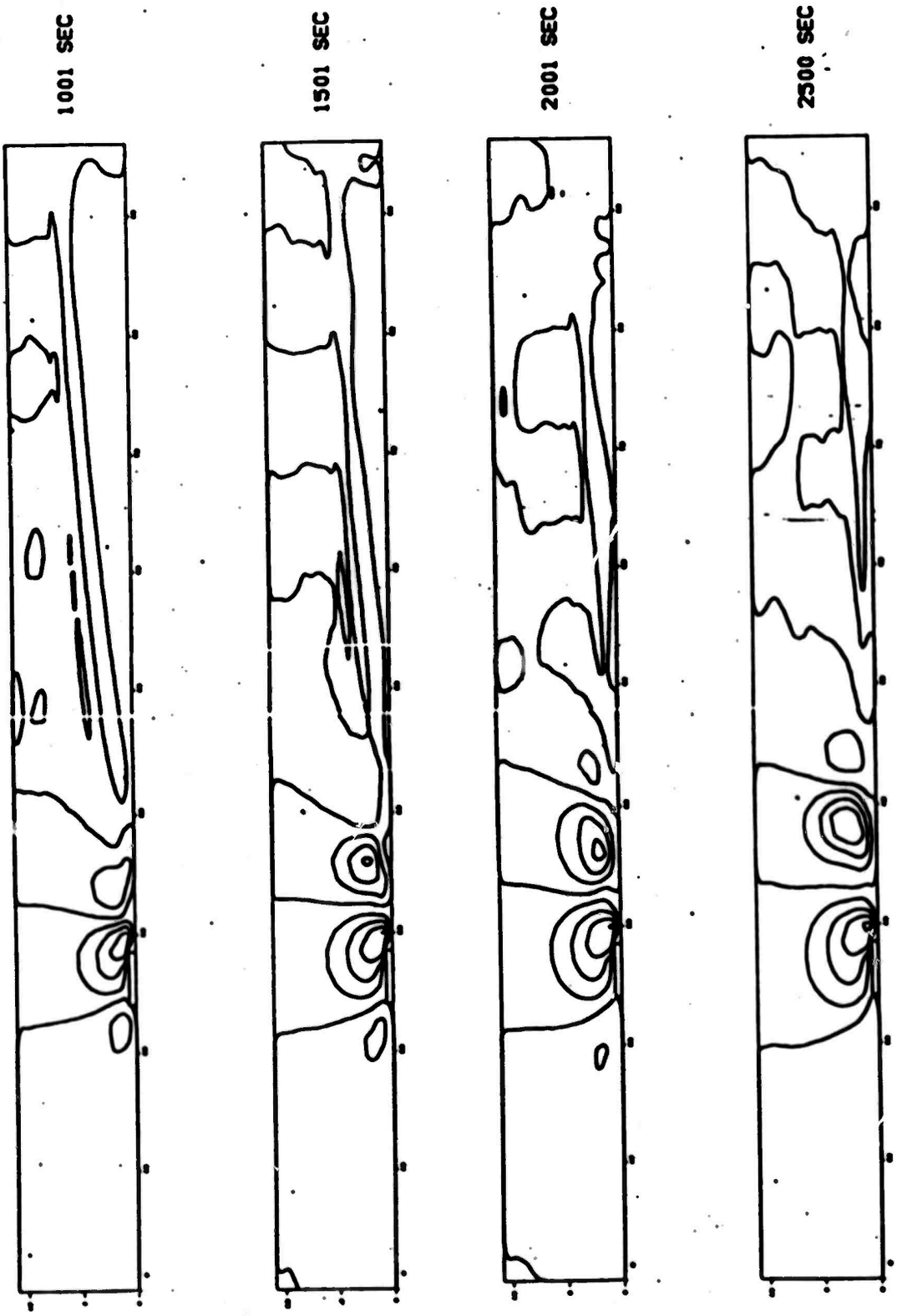


Figure 4.6 – Vertical Velocity Field from Single Wave Problem.

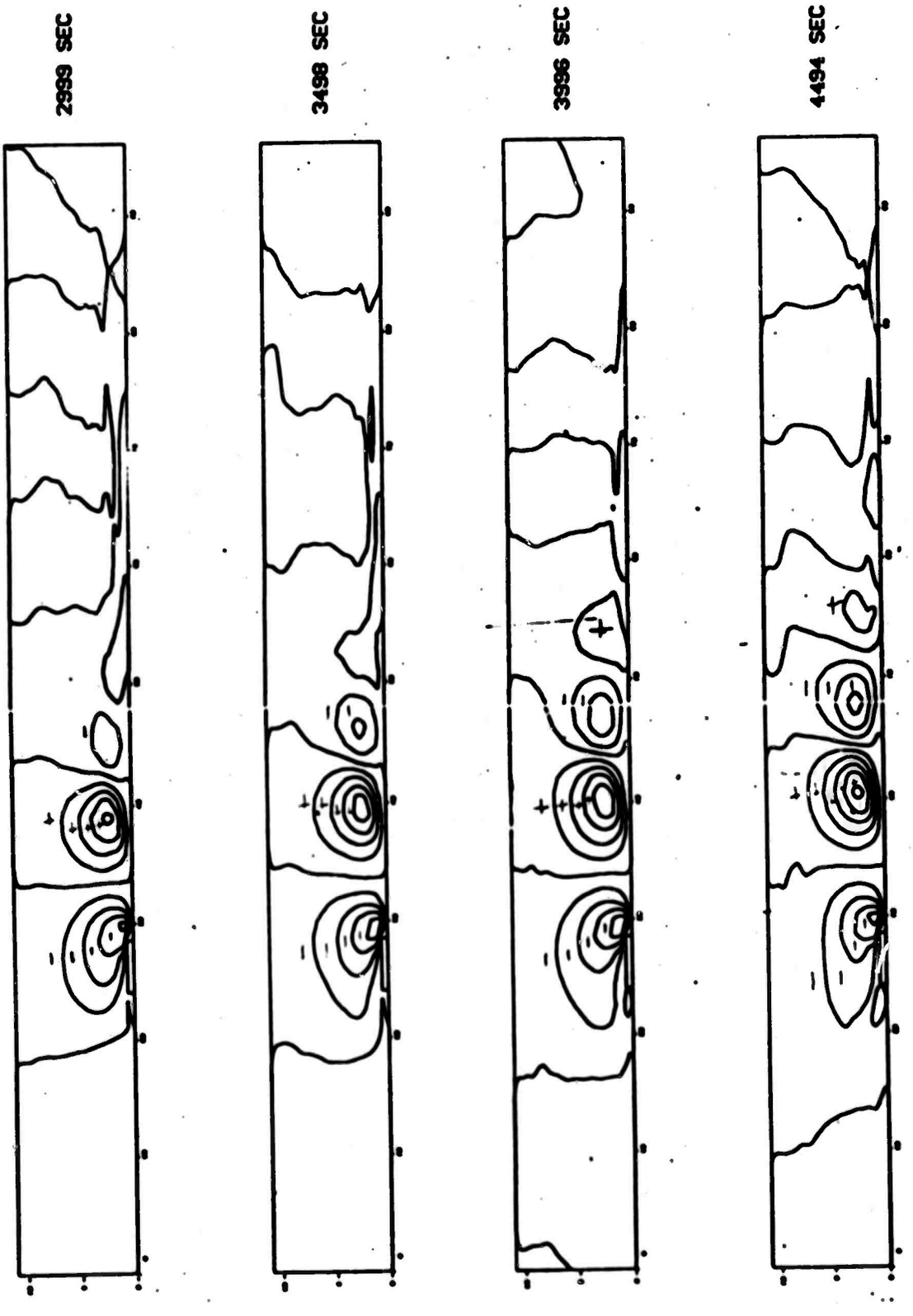


Figure 4.7 – Vertical Velocity Field from Single Wave Problem (page 2).

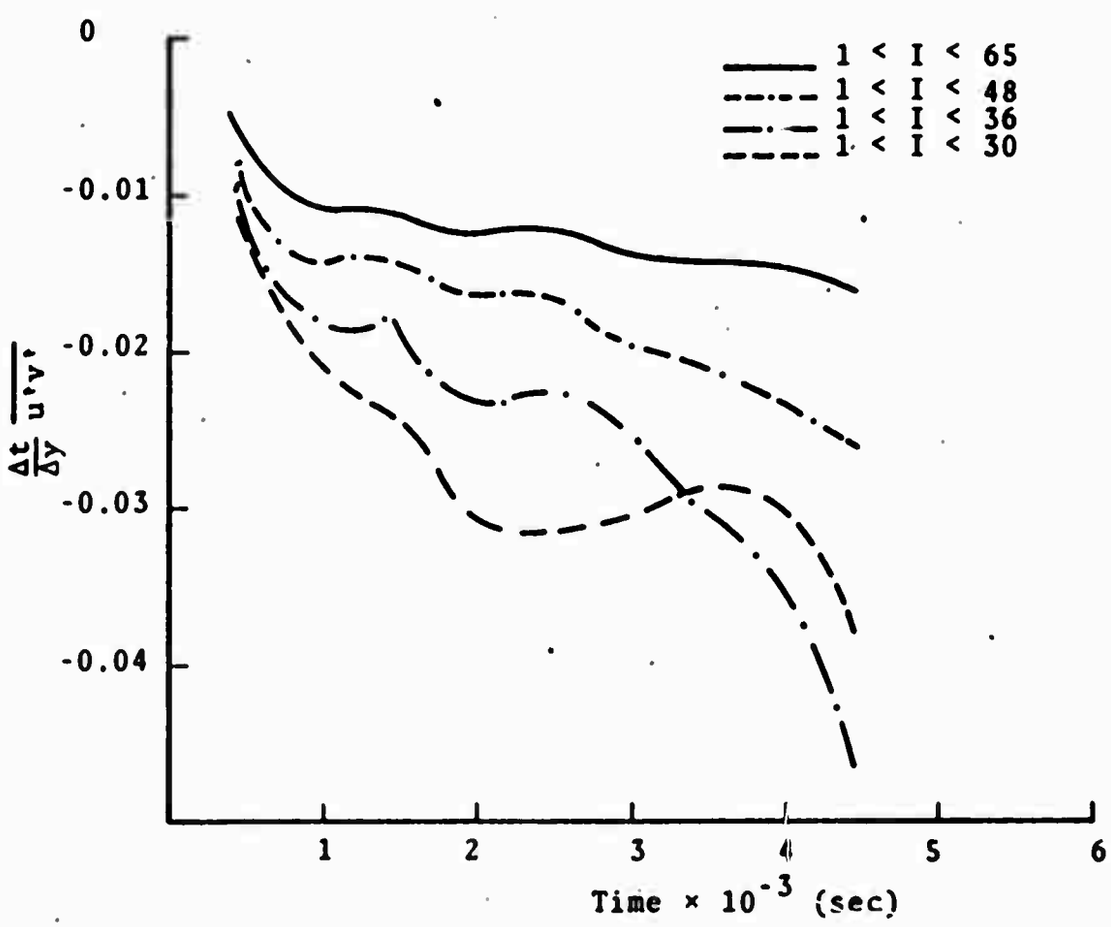


Figure 4.8 – Single Wave Problem Momentum Flux Edits.

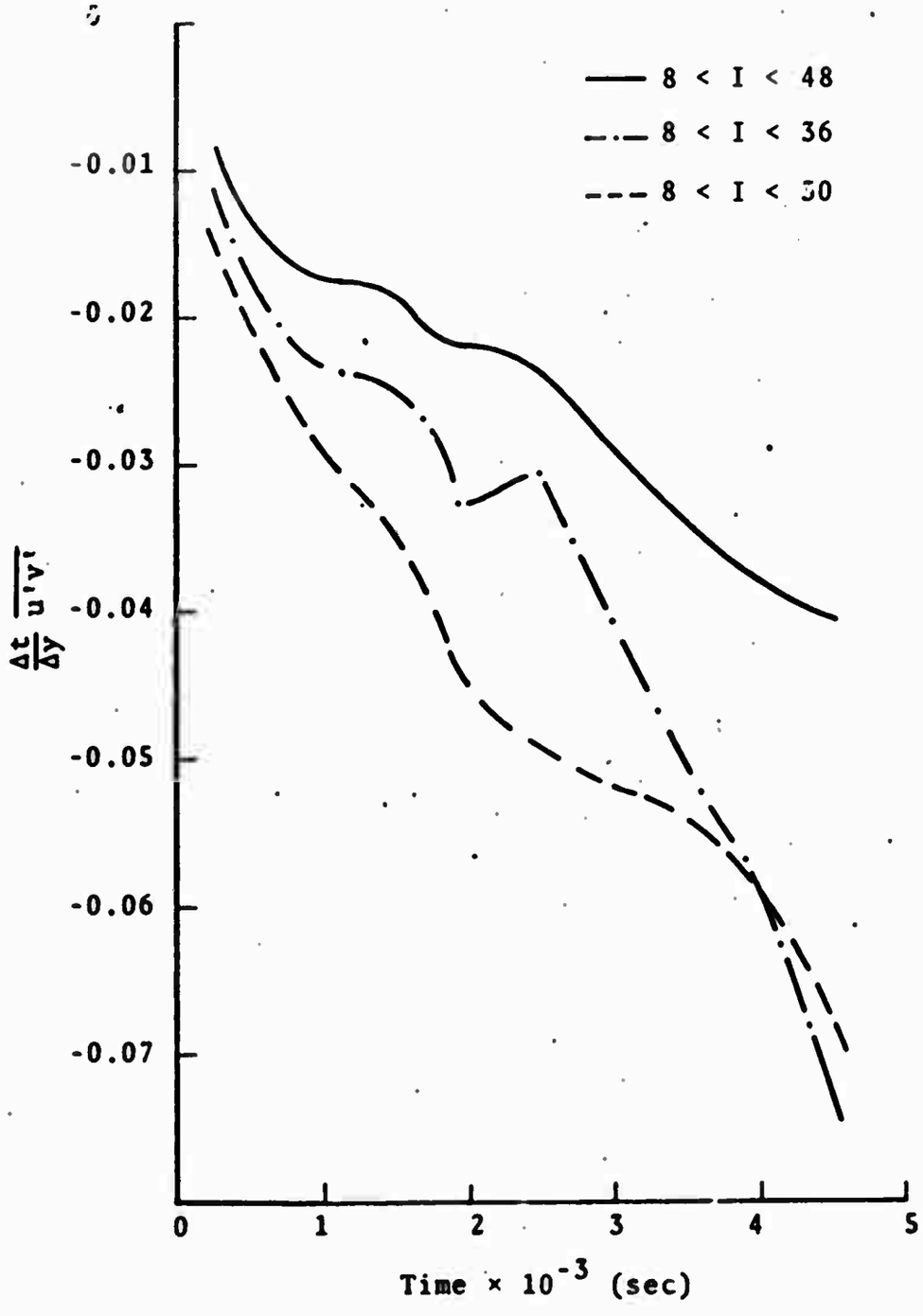


Figure 4.9 – Single Wave Problem Momentum Flux Edits (page 2).

of two of the curves occurring at approximately 3400 to 4000 seconds on either of the figures. The same phenomenon occurs when the averaging length is reduced by discounting zones from in front of the obstacle as well as the rear. While it is not clear what the averaging length should be in these cases or the intrepretation of these results, it is clear that the magnitude of the edited quantity is only equal to the drag on the mountain if the inlet and outlet values of p and ρu^2 are identical. Since this is the case only when the total numerical grid length is used as the averaging length, due to the cyclic boundary conditions, a value for the drag on the mountain can only be estimated from the uppermost curve of Figure 4.8. The value of the drag reached at 4,445 seconds was approximately equal to 10 dynes/cm^2. This value agrees qualitatively with measured values of the momentum flux reported by D. K. Lilly[12] for measurements at Boulder, Colorado.

One other important feature of the momentum flux edit is the oscillatory character of the values with time. This is thought to be related to the formation of the individual vertical velocity cells, i.e., as a new positive or negative cell is formed, the effect seems to be to increase or decrease the horizontal average of the vertical flux of horizontal momentum. This cyclic character is perhaps more clearly seen in the edits of the two wave problem discussed later.

Figure 4.10 shows the momentum edit as a function of height at a time of 4,445 seconds. The value goes to zero very quickly above the mountain. This indicates the solution is not yet approaching a steady state value since the drag for a steady problem would be constant with height.

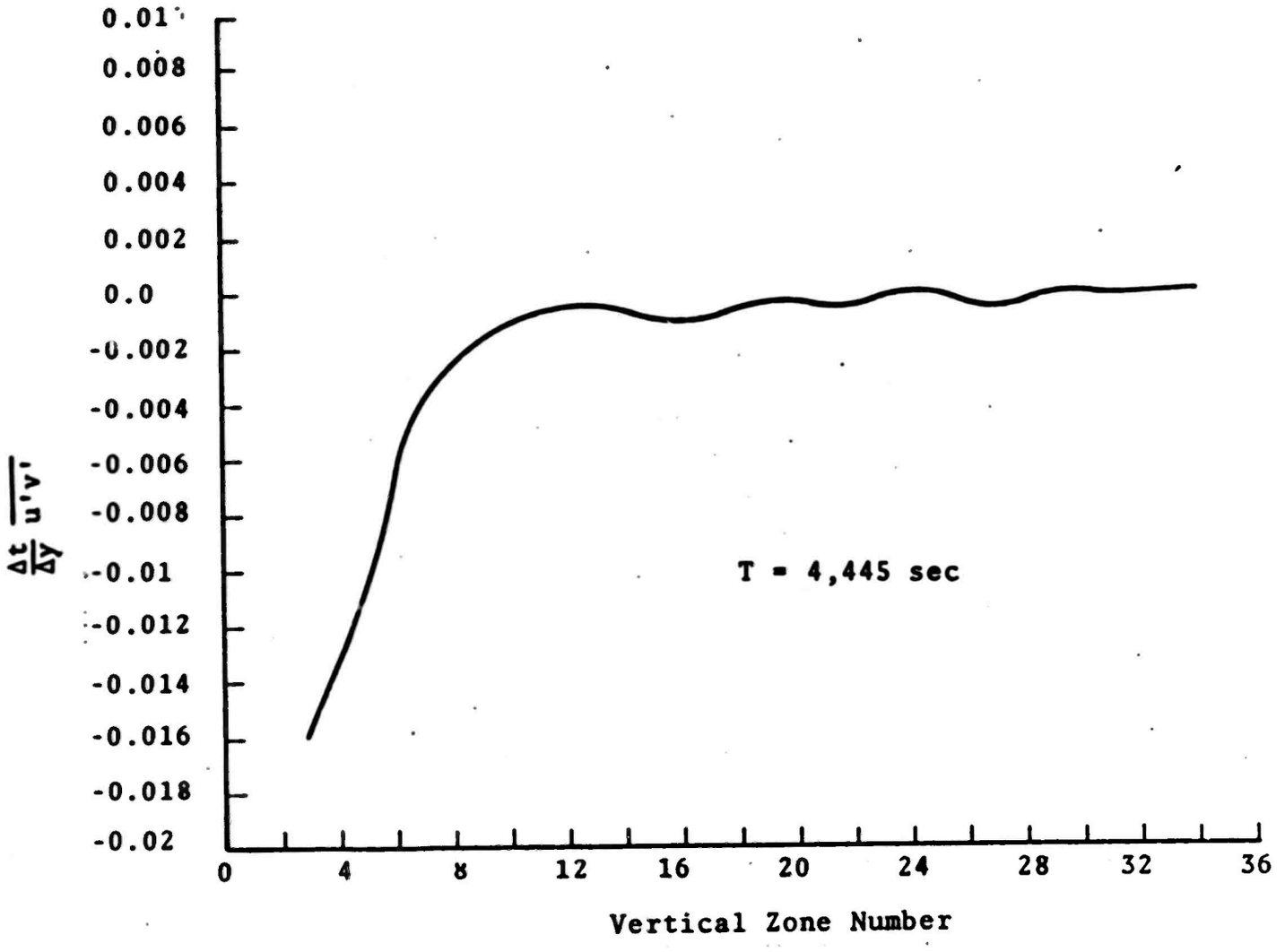


Figure 4.10 – Momentum Flux as a Function of Vertical Height.

4.2 TWO WAVE PROBLEM

The conditions for the two wave problem closely match those of a problem described by Wurtele and Foldvik.[13] These authors computed numerically the transient formation of a mountain lee wave. One wave of length 10 to 15 km was formed using conditions similar to those described for the two wave problem in Table I. Using the same type of analysis as described previously for the single wave problem, the Palm and Foldvik results indicated a wave of approximately 9.2 km wavelength as well as a second wave of approximately 25 km wavelength should be present. Private communication with Wurtele indicated that the longer wave was not noticed in their calculation. These data were incorporated into HAIFA and run to a time of 5,474 seconds. The results of this numerical calculation are presented in Figure 4.11 through 4.14. The streamlines shown in Figure 4.12 at a time of 5,474 seconds display the presence of two waves. The shorter wave appears just above and behind the obstacle and agrees with theory in that the wave length is approximately 12 km in length. A second wave appears behind the obstacle at a height of 7 to 8 km with a wavelength of approximately 22 km. The small discrepancies between predicted and calculated wavelengths are probably the result of the linear theory used in producing the diagram of Palm and Foldvik. The presence of the second wave is also strongly evident in the bending over of the vertical velocity fields. The interaction of the two waves is seen by the presence of small vertical velocity regions whose direction is opposite of the velocity cell completely surrounding it.

The figures showing the growth of the vertical velocity cells may be compared to the results of Wurtele and Foldvik. The Figure 4.15 was taken from their paper and indicates

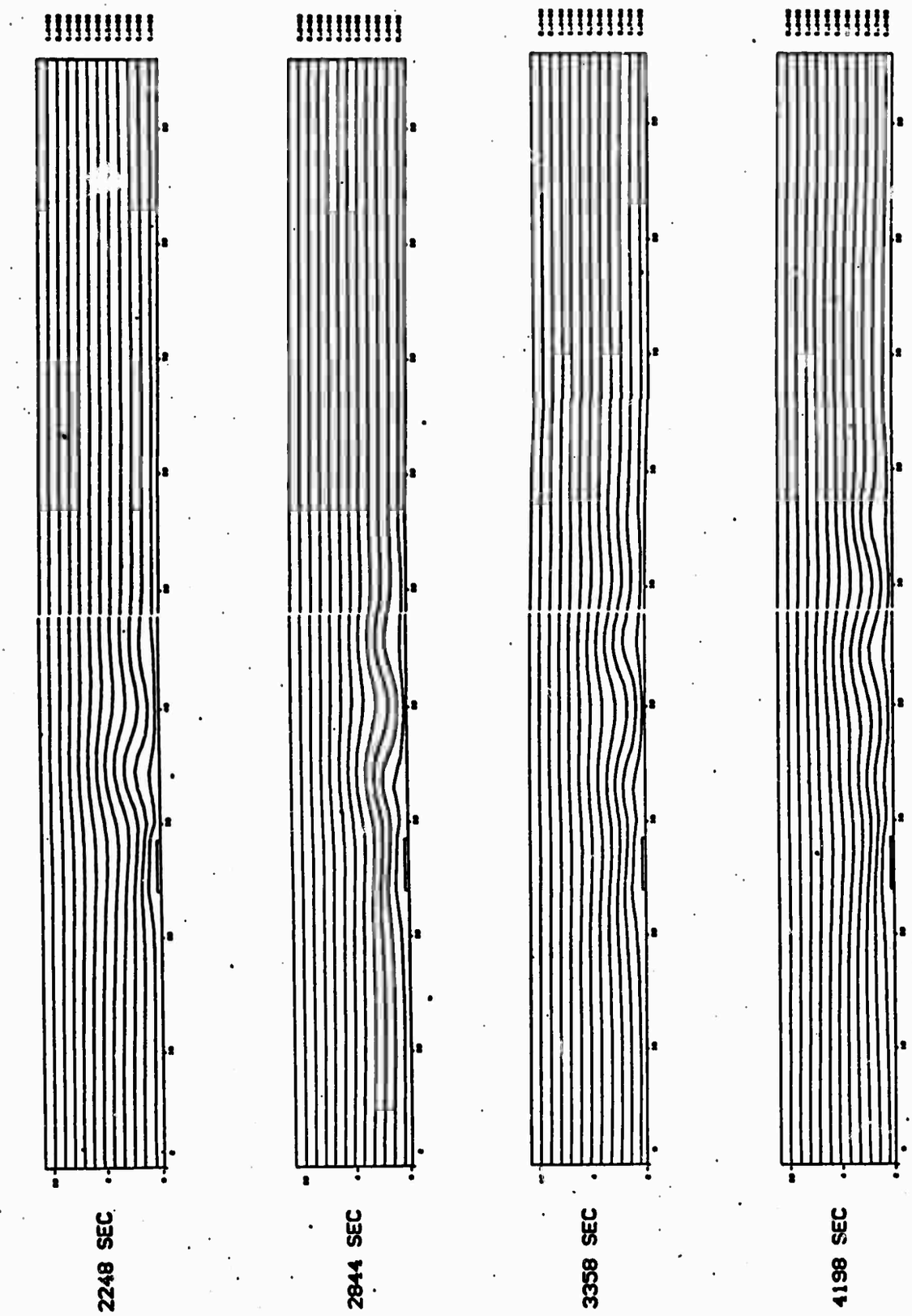


Figure 4.11 — Streamlines from Two Wave Problem.

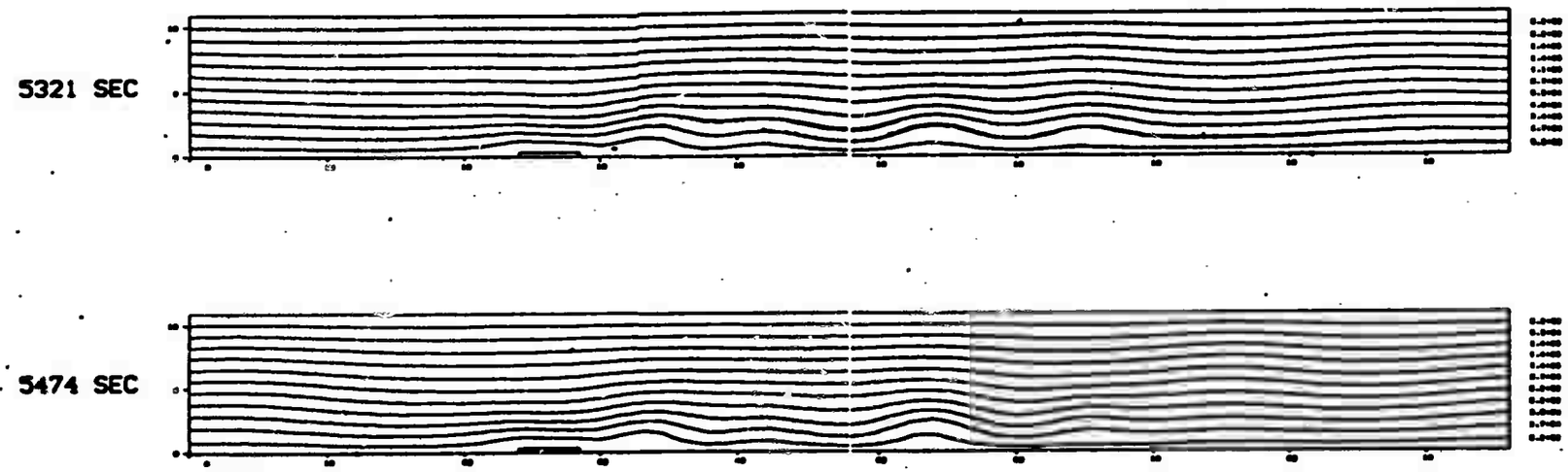


Figure 4.12 – Streamlines from Two Wave Problem (page 2).

Figure 4.13 – Vertical Velocity Field from Two Wave Problem.

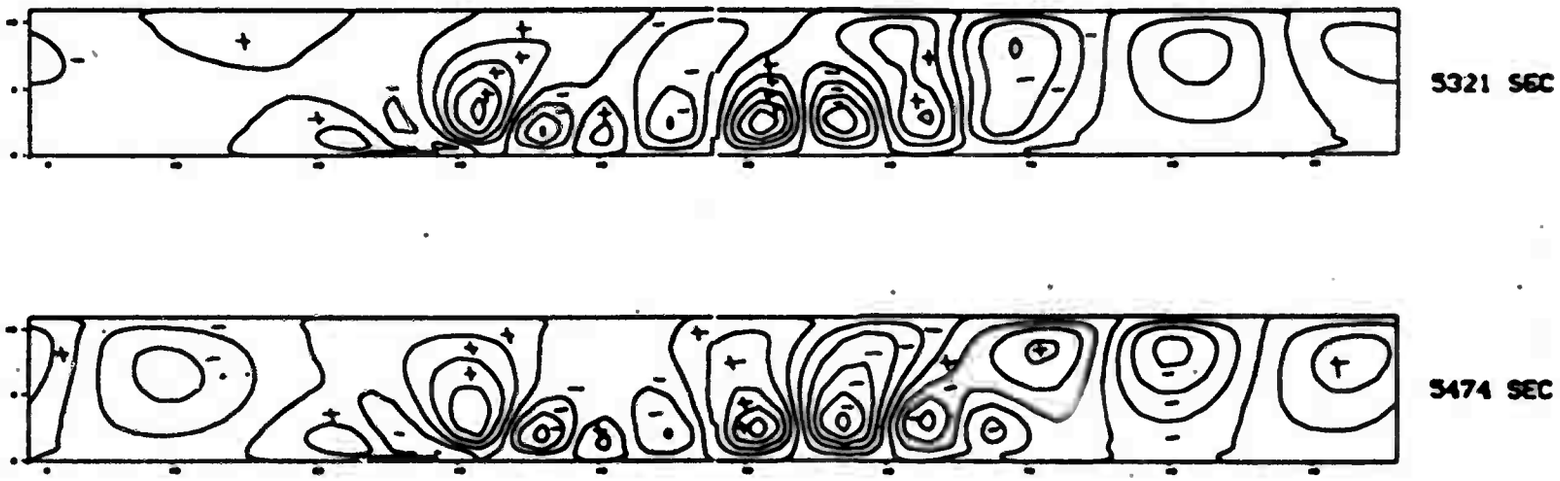


Figure 4.14 – Vertical Velocity Field from Two Wave Problem (page 2).

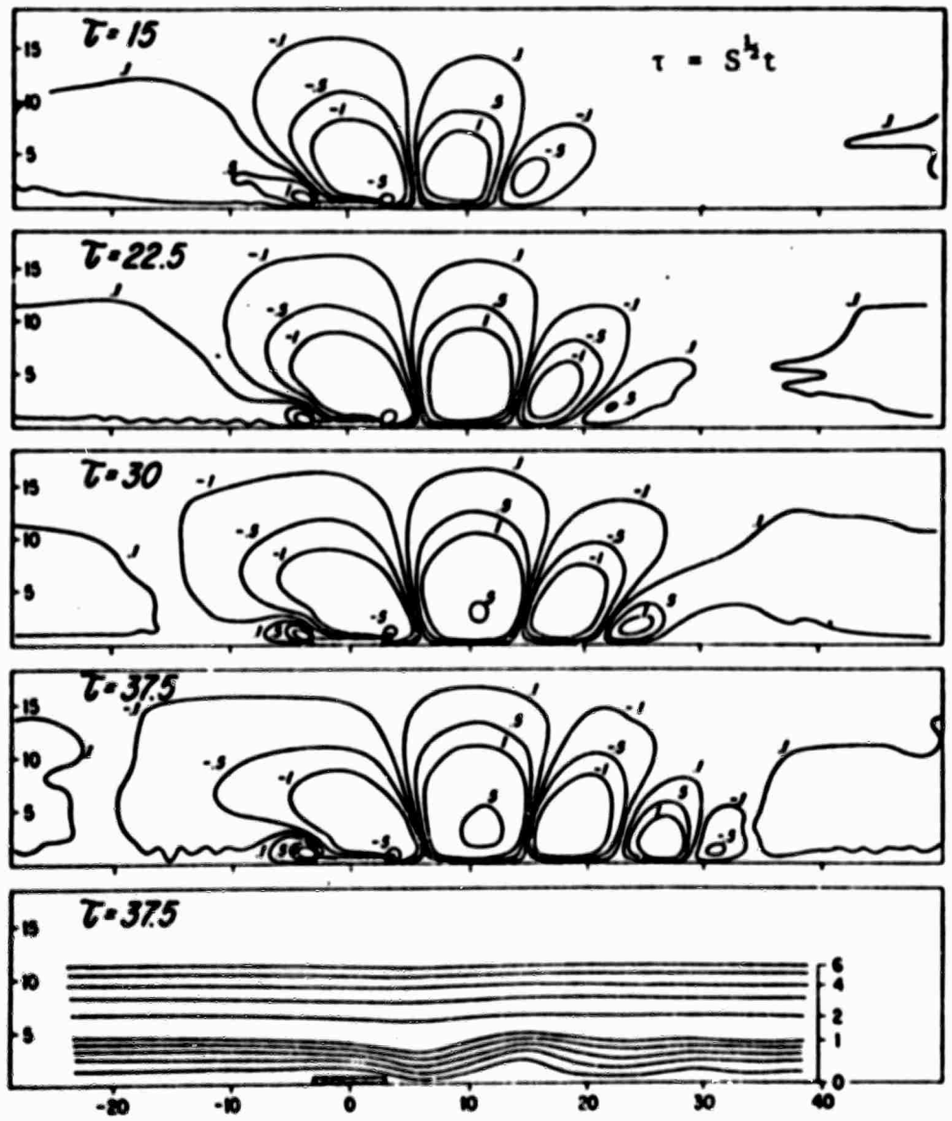


Figure 4.15 – The field of vertical velocity as calculated by Wurtele and Foldvik. The lowest panel represents the total streamline field.

a large negative vertical velocity cell over the obstacle for all time. The positive cell just downstream of the obstacle is moving toward the upstream side of the obstacle with time but does not ever dominate the flow over the obstacle. Our calculations indicate a negative cell over the obstacle for times to 5,300 seconds. Shortly after that time, the large positive cell at the rear of the obstacle combines with the small positive cell forward of the obstacle. A small negative cell still remains over the obstacle itself, but there is no longer a dominate negative or downwind flow over the obstacle. These differences are not well understood. The phenomena may be due to differences in initial or boundary conditions as none of these were completely indicated in the Wurtele and Foldvik paper. The additional time of the computation or the presence of the second wave may also be responsible for this phenomena. The length of the obstacle, which was not specified in the previous study, was found in HAIFA calculations to have a definite influence on the pattern of vertical velocities over the obstacle. The temperature distribution, which was specified as equal to one-half the dry adiabatic, may also vary a small amount. Any of these parameters may have caused the small differences seen in our calculation and that reported by Wurtele and Foldvik. The similarities, particularly of the vertical velocity cells at times less than 5,000 seconds, certainly indicate good qualitative agreement.

The momentum edit of the two wave problem, shown in Figure 4.16, indicates a much more cyclic nature than of the single wave problem. Each half cycle appears to have some correspondence to the complete formation of a vertical cell although the interaction of the two waves present makes the intrepretation of the data difficult.

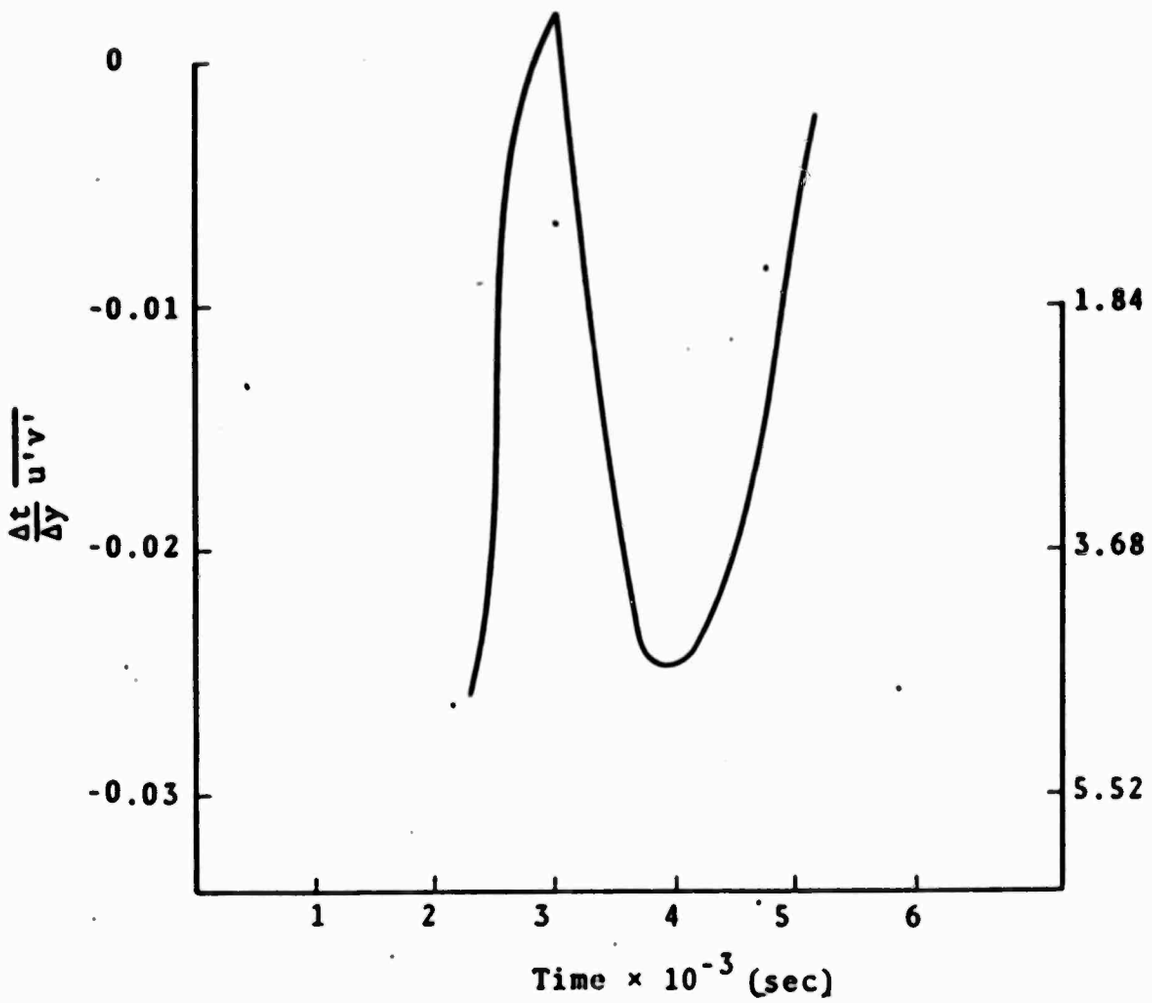


Figure 4.16 – Momentum Flux Edit from Two Wave Problem.

4.3 UNIFORM VELOCITY

A problem using a velocity distribution uniform with height and equal to 10 m/sec perturbed by a one kilometer high mountain was completed. The lapse rate was set equal to one-half the dry adiabatic. Figures 4.17 through 4.20 show the resulting streamlines and vertical velocity cells formed under these conditions Wurtele and Foldvik have also investigated this problem, the results of which are shown in Figures 4.21 through 4.23. A comparison of their streamlines with our results show a continuous spectrum of waves is excited in both calculations, which when added together produce growing numbers of upwind-tilting troughs and crests extending to great heights. The figures showing the vertical velocity cells at the forward and rear of the obstacle show these upwind-tilting troughs and crests even more distinctly. Lyra[14] theoretically showed these same results using a linear analysis. His steady state analytical result for the streamlines and the vertical velocity field are shown in Figures 4.24 and 4.25. While there are certainly similarities in the results of Lyra, Wurtele and Foldvik, and the S^3 calculation, there are also some significant differences. The four total streamline fields computed by Wurtele and shown in Figures 4.21 and 4.22 show a large amplitude wave just above the lee slope. The vertical velocity in this region is more than five times the upstream wind and the total horizontal velocity is negative at some grid points. This feature is not present in the linear theory and did not appear in the S^3 computations. We are continuing to investigate these differences.

One of the most significant items found in calculating this problem was a numerical instability associated with the flow when the normal stability criteria for the advective terms of the equations was used. An initial computation

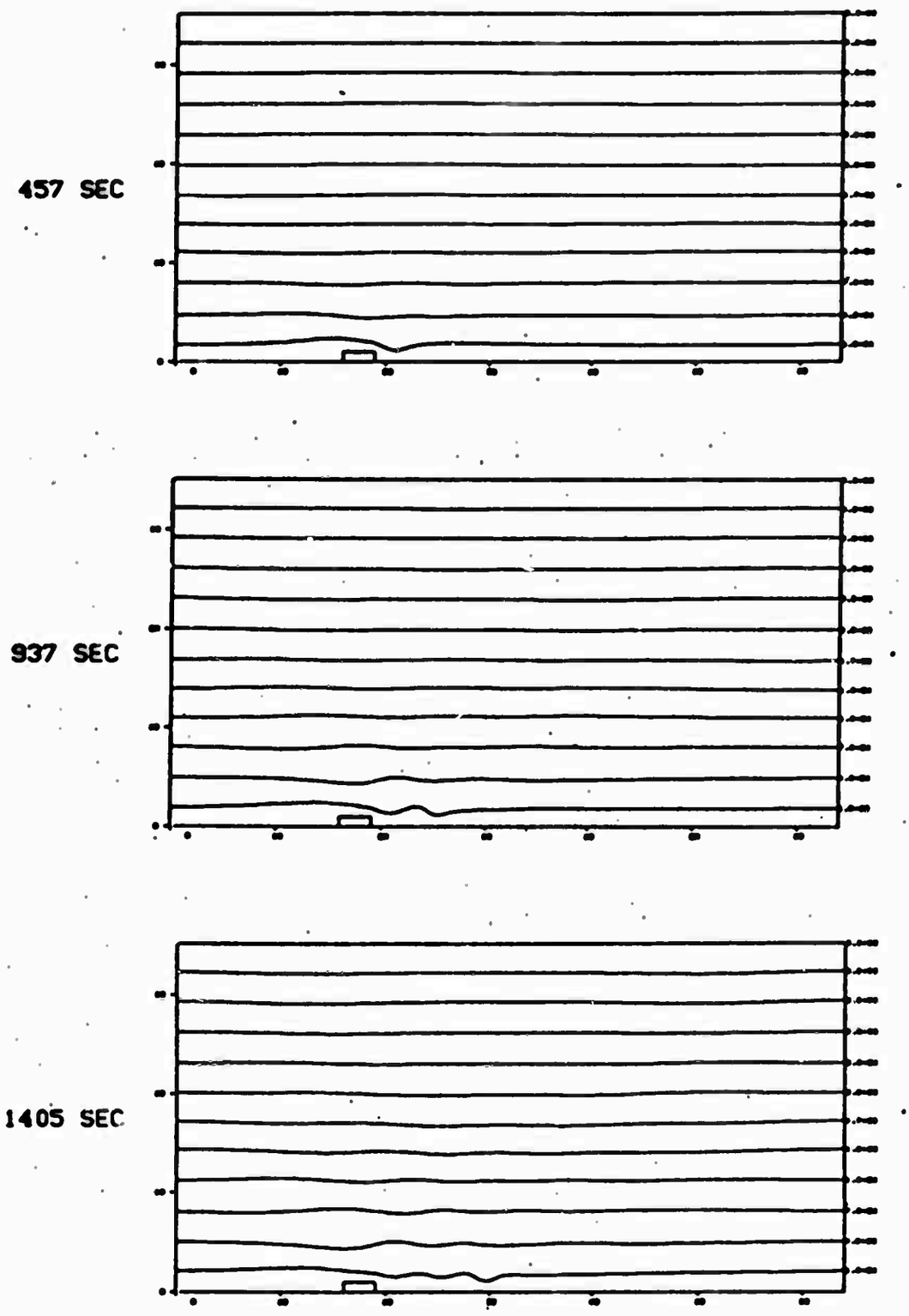


Figure 4.17 – Streamlines from Uniform Velocity Problem.

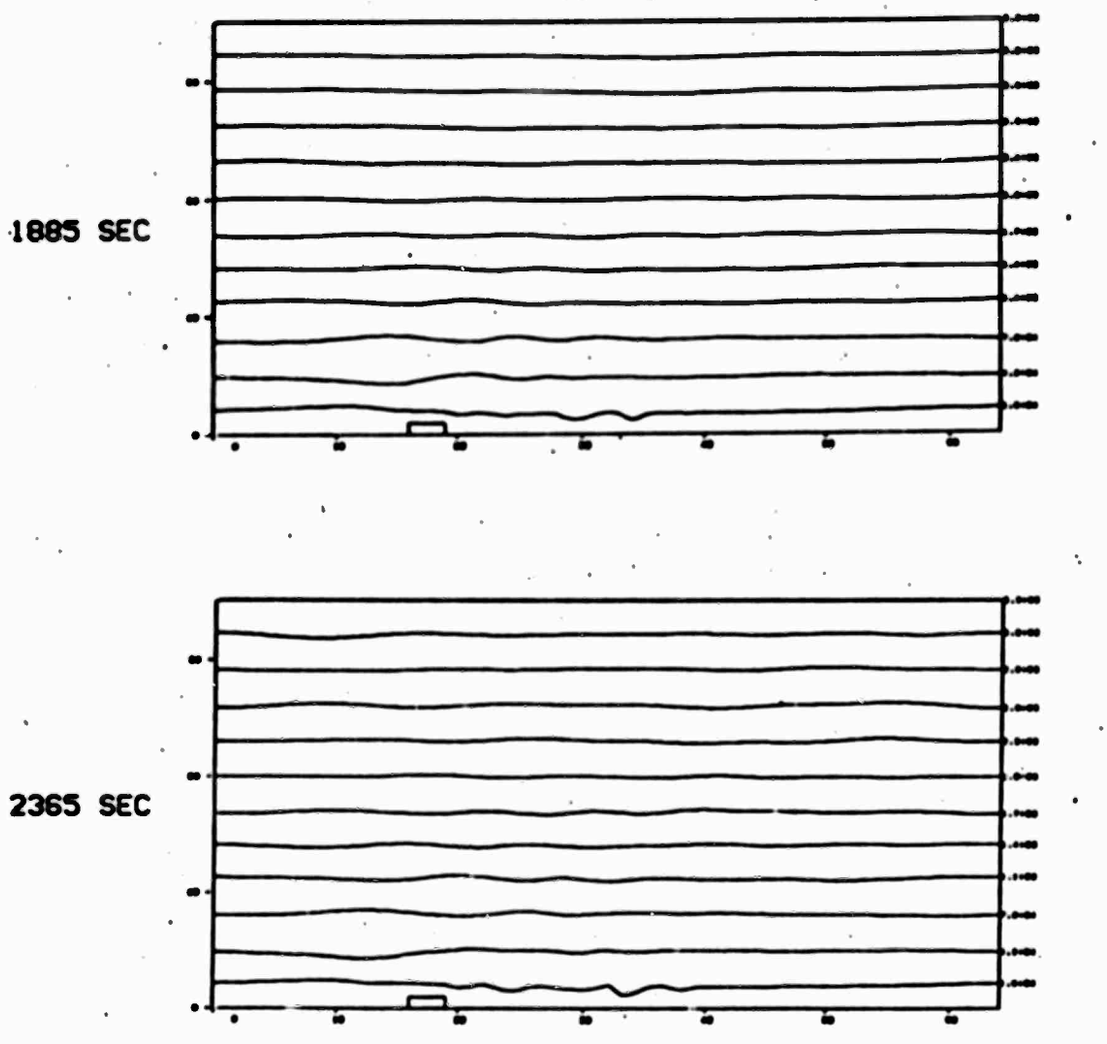


Figure 4.18 – Streamlines from Uniform Velocity Problem (page 2).

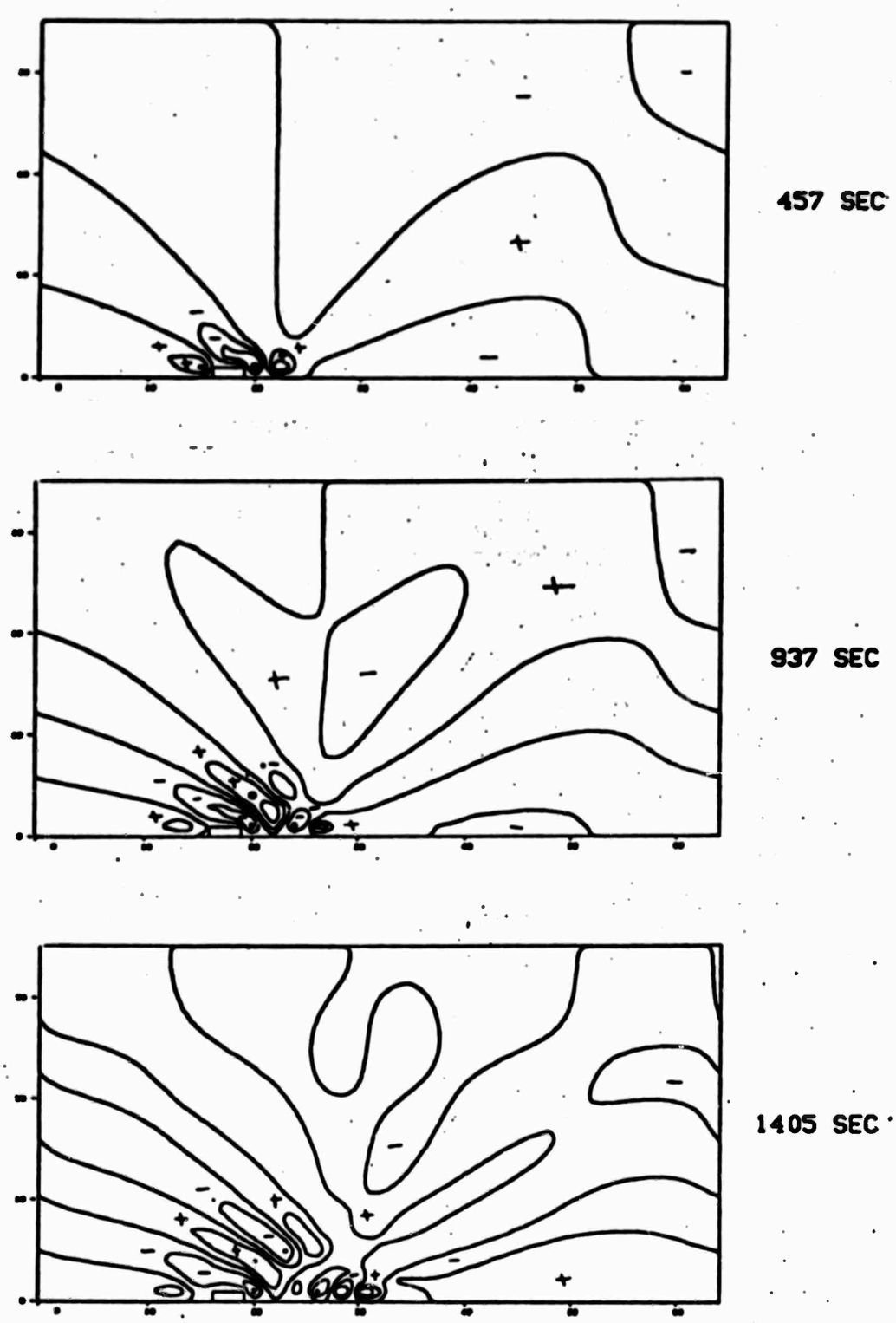


Figure 4.19 – Vertical Velocity Field from Uniform Velocity Problem.

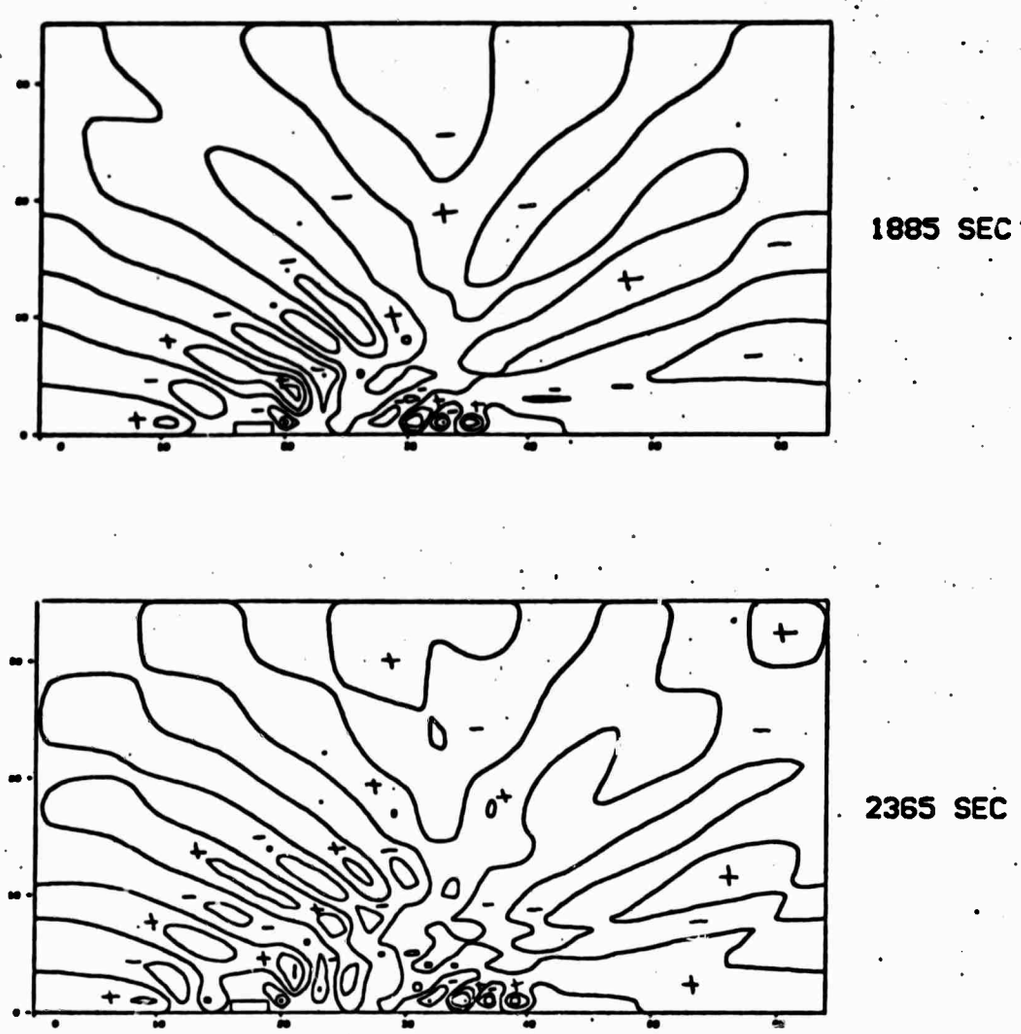


Figure 4.20 – Vertical Velocity Field from Uniform Velocity Problem (page 2).

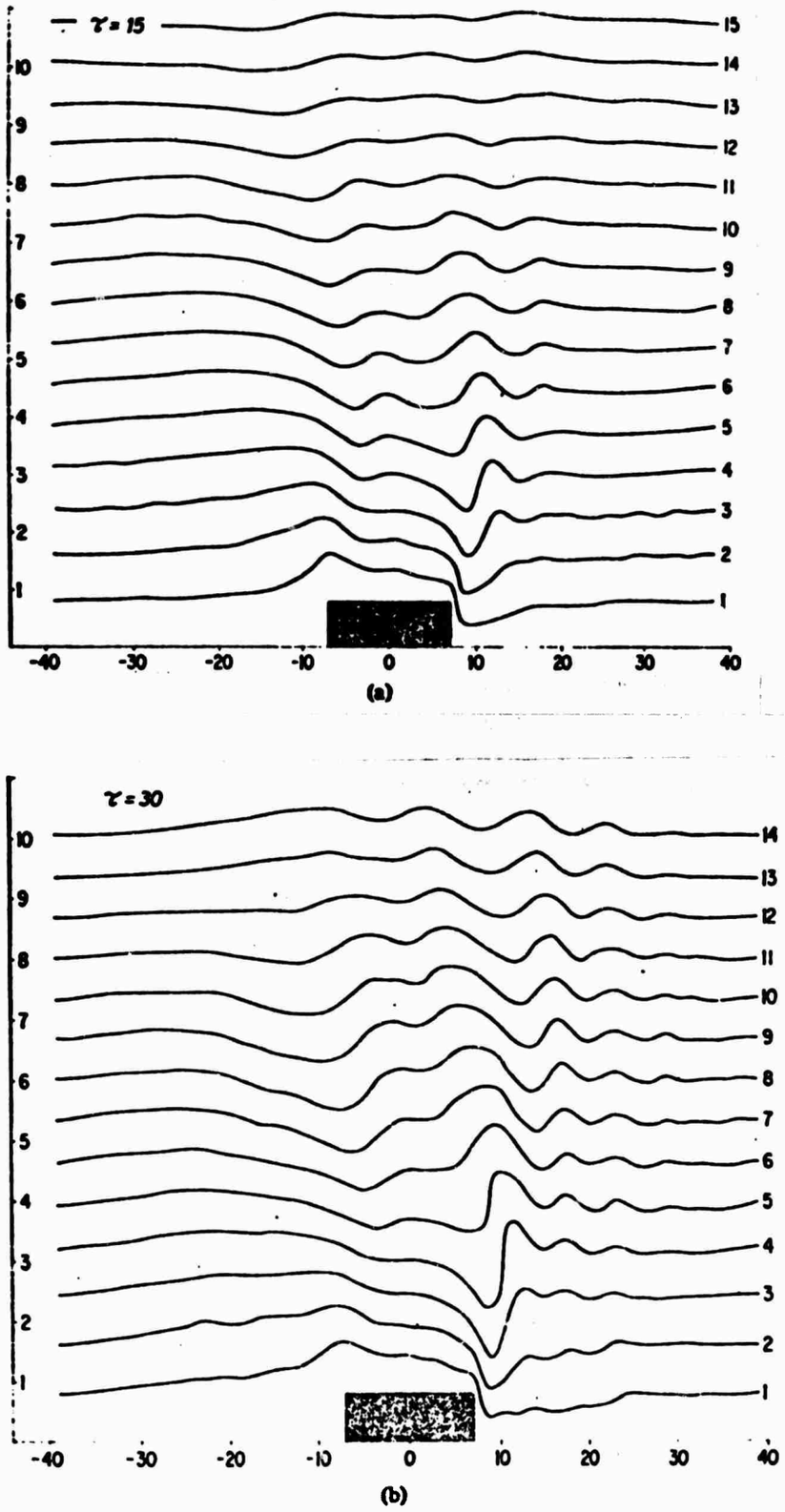
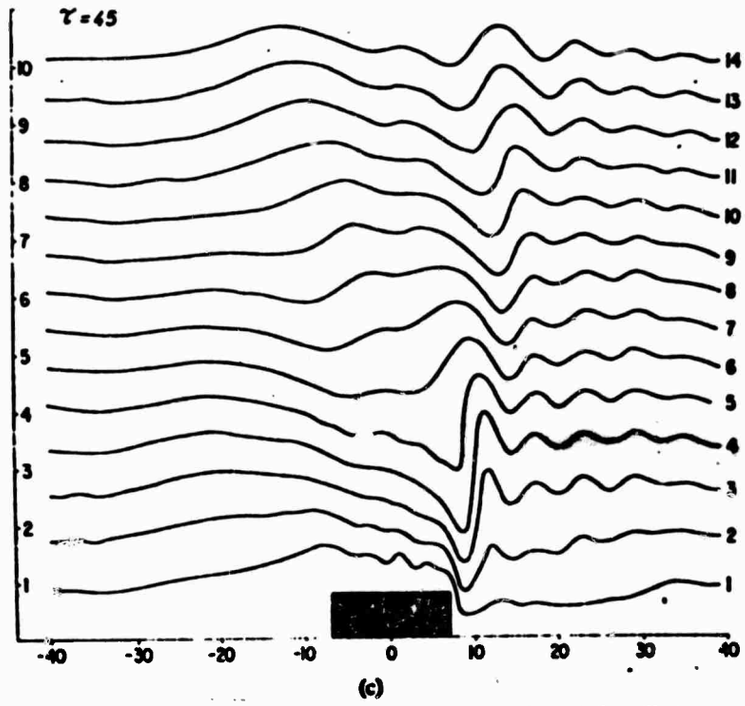
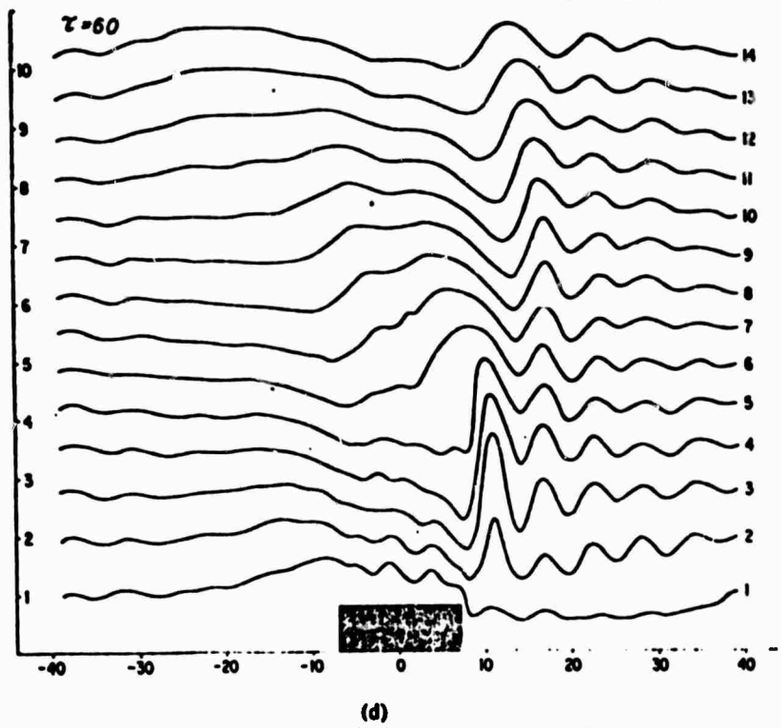


Figure 4.21 — Computed Streamlines from Wurtele and Foldvik at successive times for Uniform Velocity Problem.



(c)



(d)

Figure 4.22 – Computed Streamlines from Wurtele and Foldvik at successive times for Uniform Velocity Problem (page 2).

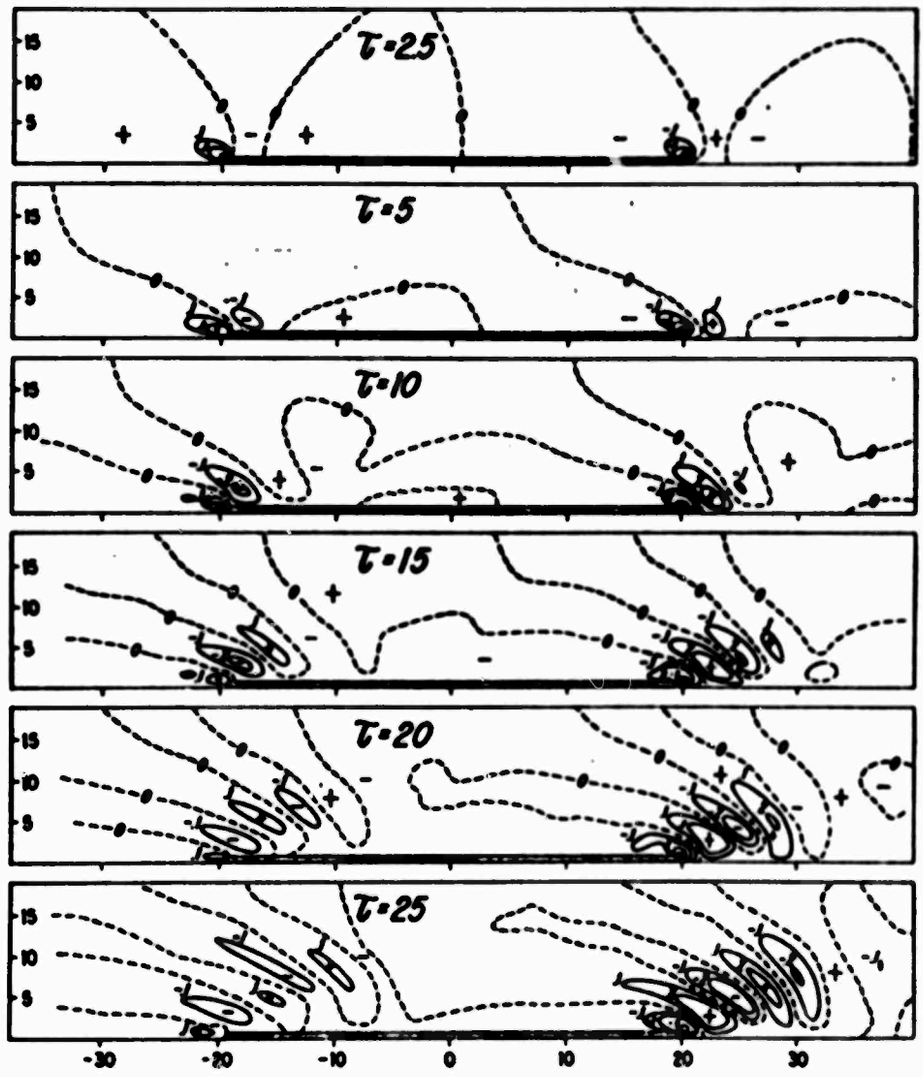


Figure 4.23 – Field of vertical motion computed under upwind conditions for successive times from Uniform velocity problem.

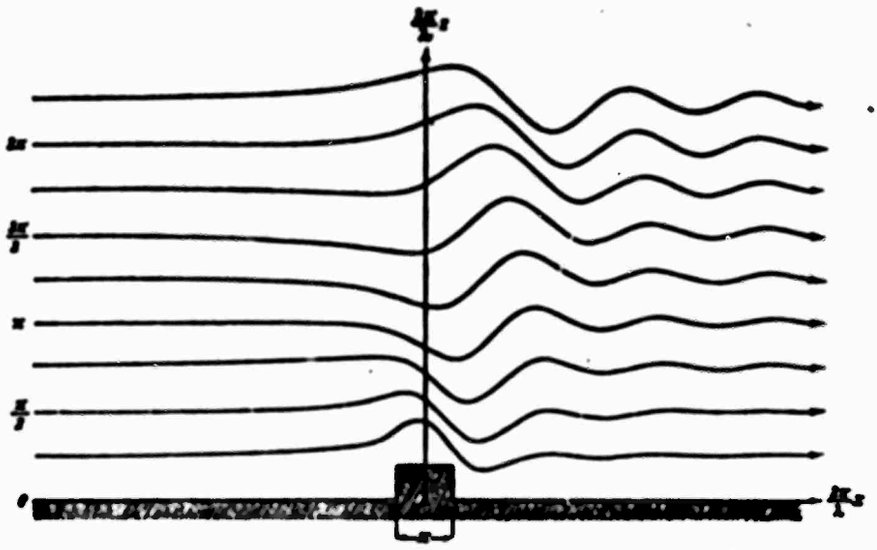


Figure 4.24 – Streamlines from the linear theory (after Lyra).

Figure4.25 – Field of vertical velocity when U = constant with height (after Lyra). Isopleths for w > 0 only.

using this time step control produced a series of large wave length high amplitude waves which propagated throughout the flow very quickly. The problem was recalculated by putting an upper limit on the time step which was based on the phase speed of the largest of these waves, i.e., a wave with a 50 km wavelength. This limited the time step to less than 14 seconds per cycle assuming the overall criteria to be that the 50 km wave would not completely traverse a grid cell in one cycle. The actual limiting time step used in the recalculation was 12.0 seconds. The resulting wave pattern is the one shown and previously discussed in this section. This new stability criteria, which had not been previously used, was not required in earlier problems due to either (1) the damping of the disturbances caused by the wind shear or (2) the high velocities in the single wave problem controlling the time step to an acceptable value. Later problems, the tropopause and inversion layers, exhibited this same instability.

4.4 INVERSION LAYER I

The determination of the effect of an inversion layer in the atmosphere was calculated using the basic HAIFA code. The inversion layer was described as a positive 4°C temperature change over a 1.5 km height as shown in Figure 4.26. The other initial conditions are described in Table I. The results, shown in Figures 4.27 through 4.30, indicate a small effect in the vertical velocity cells at heights corresponding to the inversion heights. The cells appear to be broader at a 5 km height than those seen in the two wave case for example. There also appear to be displacements in the vertical cells at this position. However, these may be due more to the change in the lapse rates at this position than the presence of the 4°C temperature increase.

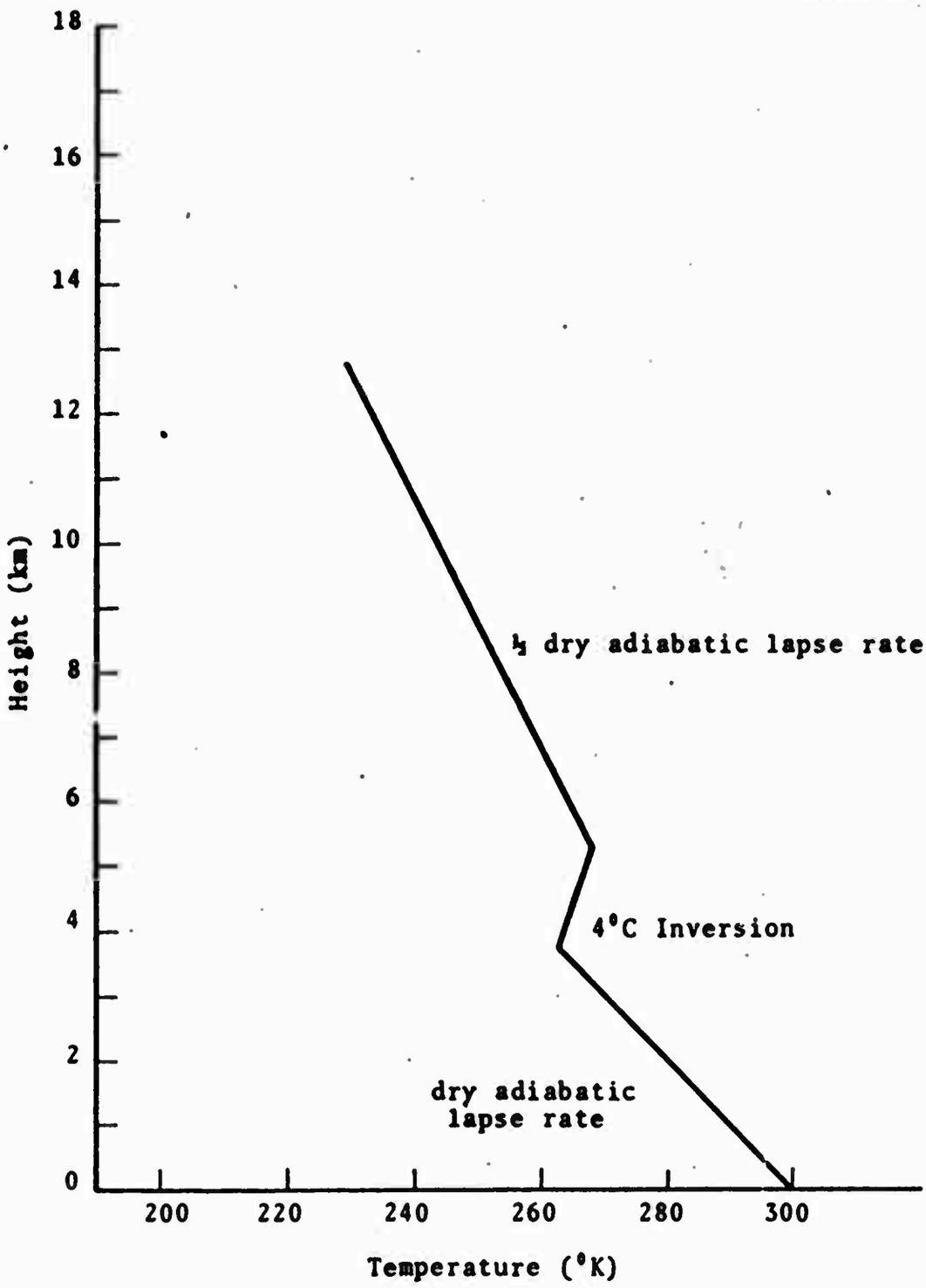
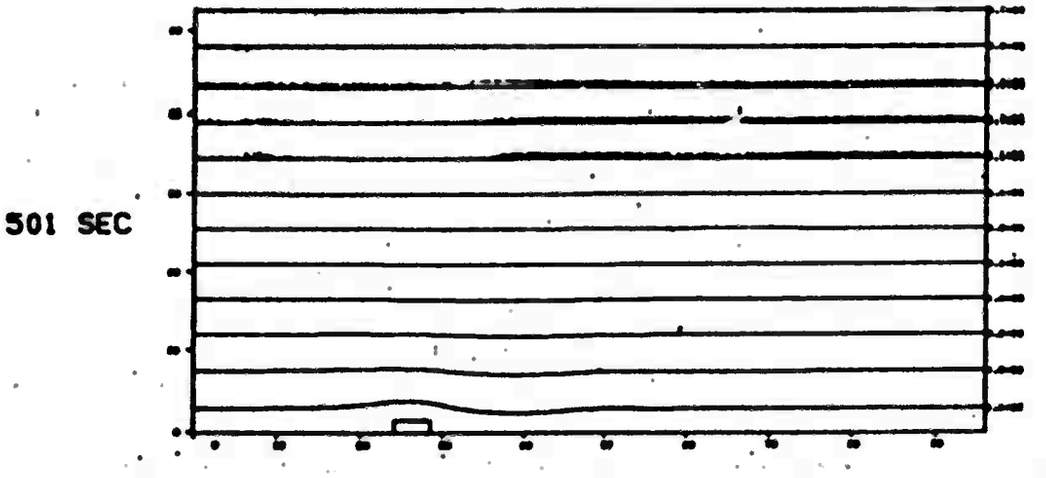


Figure 4.26 – Temperature Distribution used in Inversion Layer Problem.



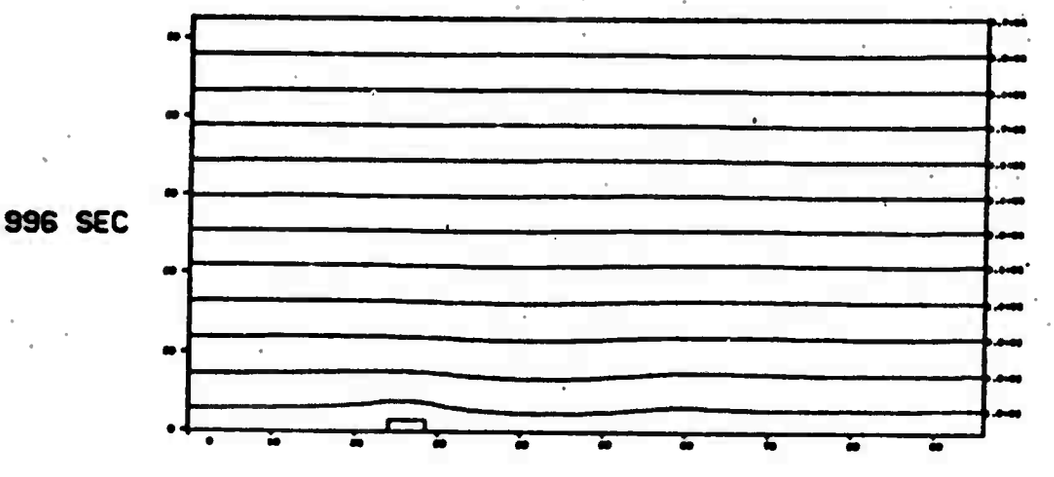


Figure 4.27 – Streamlines from Inversion Layer Problem.

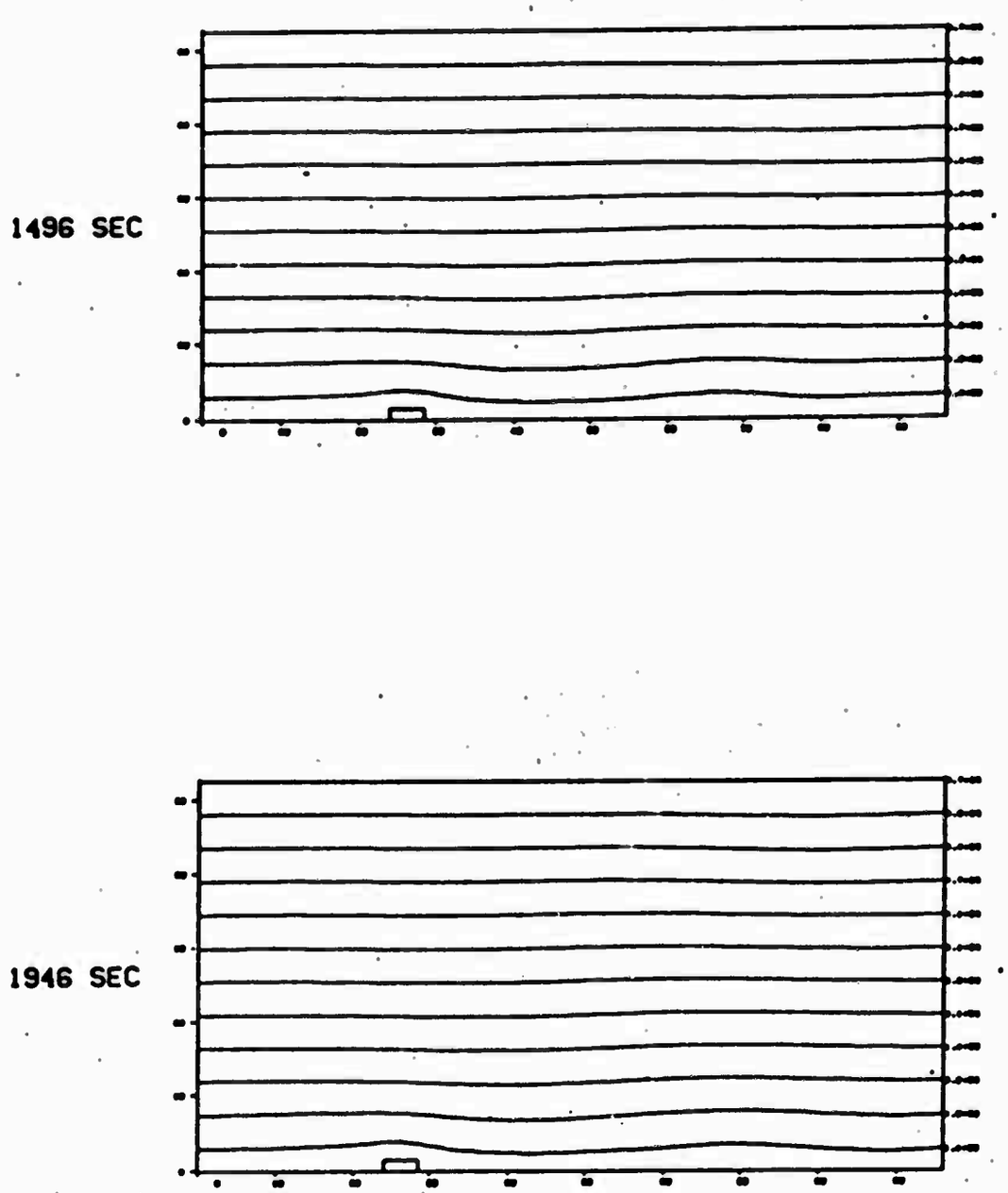
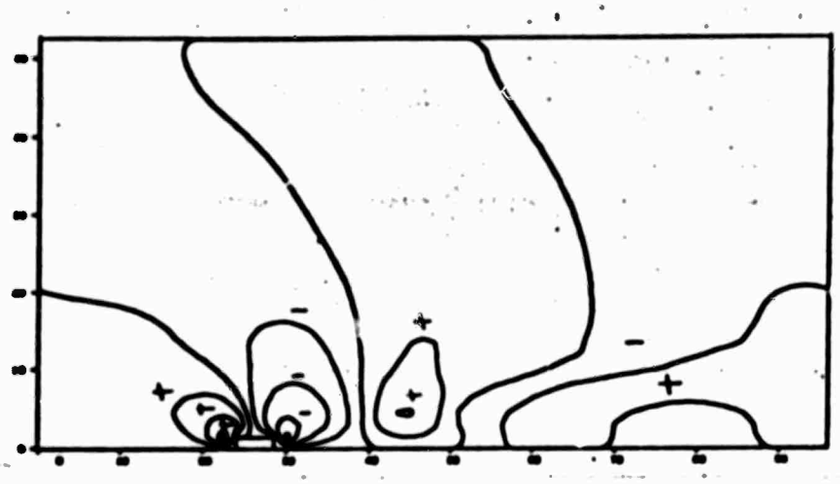


Figure 4.28 – Streamlines from Inversion Layer Problem (page 2).



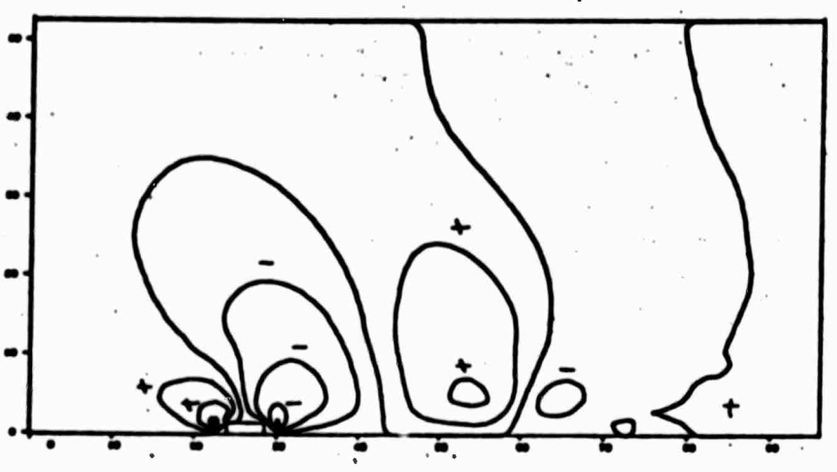
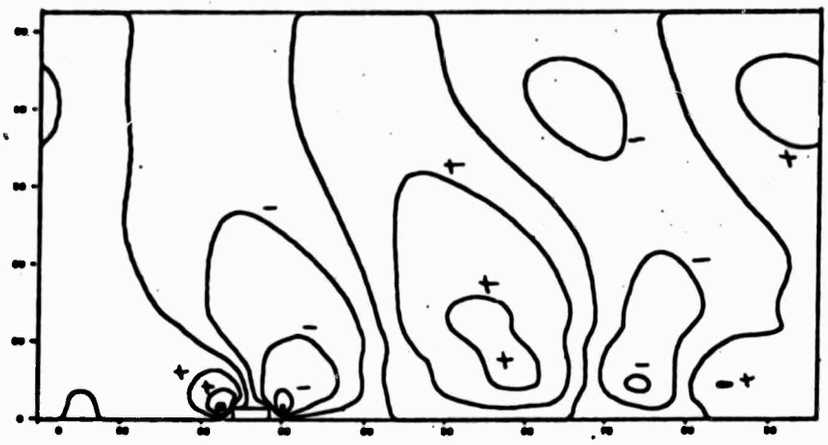
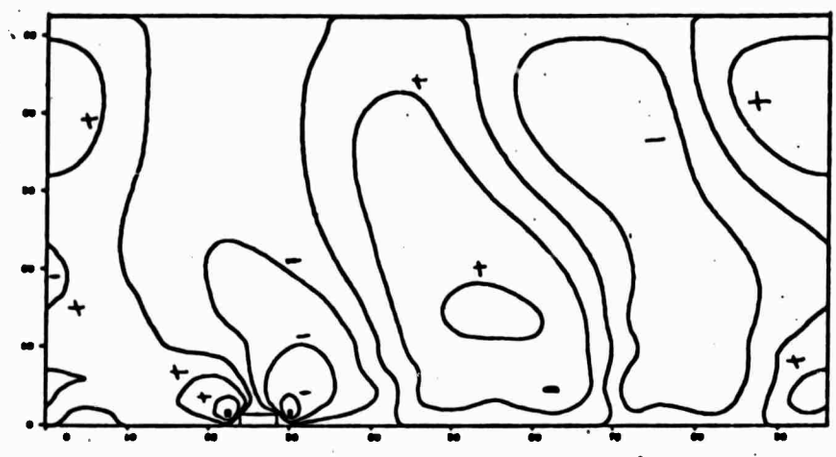


Figure 4.29 – Vertical Velocity Field from Inversion Layer Problem.



1496 SEC



1946 SEC

Figure 4.30 – Vertical Velocity Field from Inversion Layer Problem (page 2).

Because of the coarse zoning at the inversion layer, the definition of the flow is poor. This problem will be recalculated using the variable zoning code and the results will be reported in the final report of the contract.

4.5 TROPOPAUSE PROBLEM

The test calculation representing a tropopause problem consisted of initial conditions as described in Table I. The calculated streamlines and vertical velocity contours are shown in Figures 4.31 through 4.34. The most noticeable characteristic of the resulting solution is the tilting of the vertical velocity cells toward the upwind direction. The streamline pattern for this problem did indicate, but not clearly, this same phenomena of the upwind tilting of the gravity wave peaks.

Figure 4.47

87 SEC

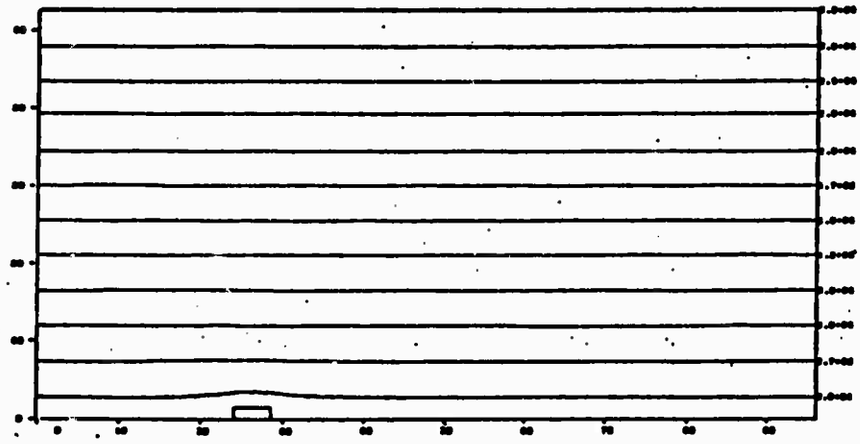


Figure 4.48

287 SEC

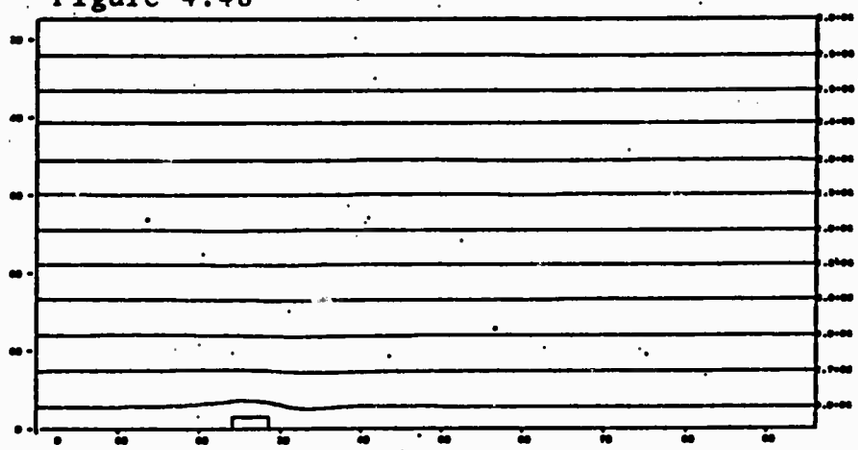


Figure 4.49

487 SEC

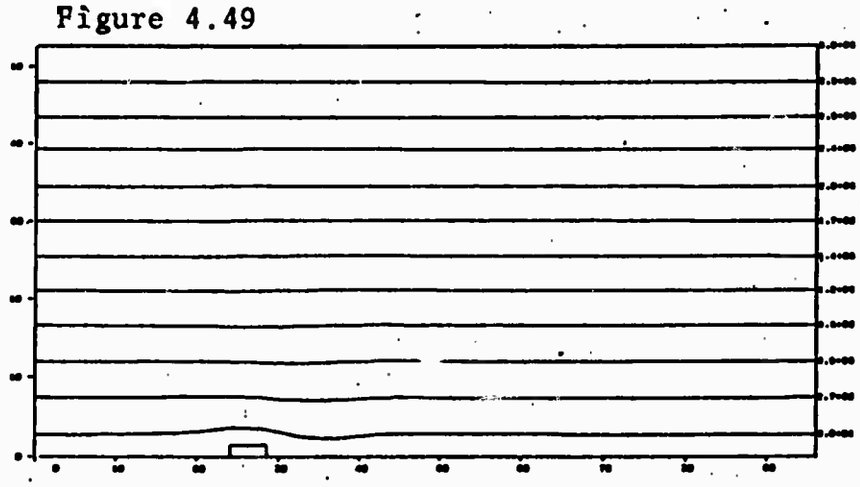


Figure 4.31 – Streamlines from Tropopause Problem.

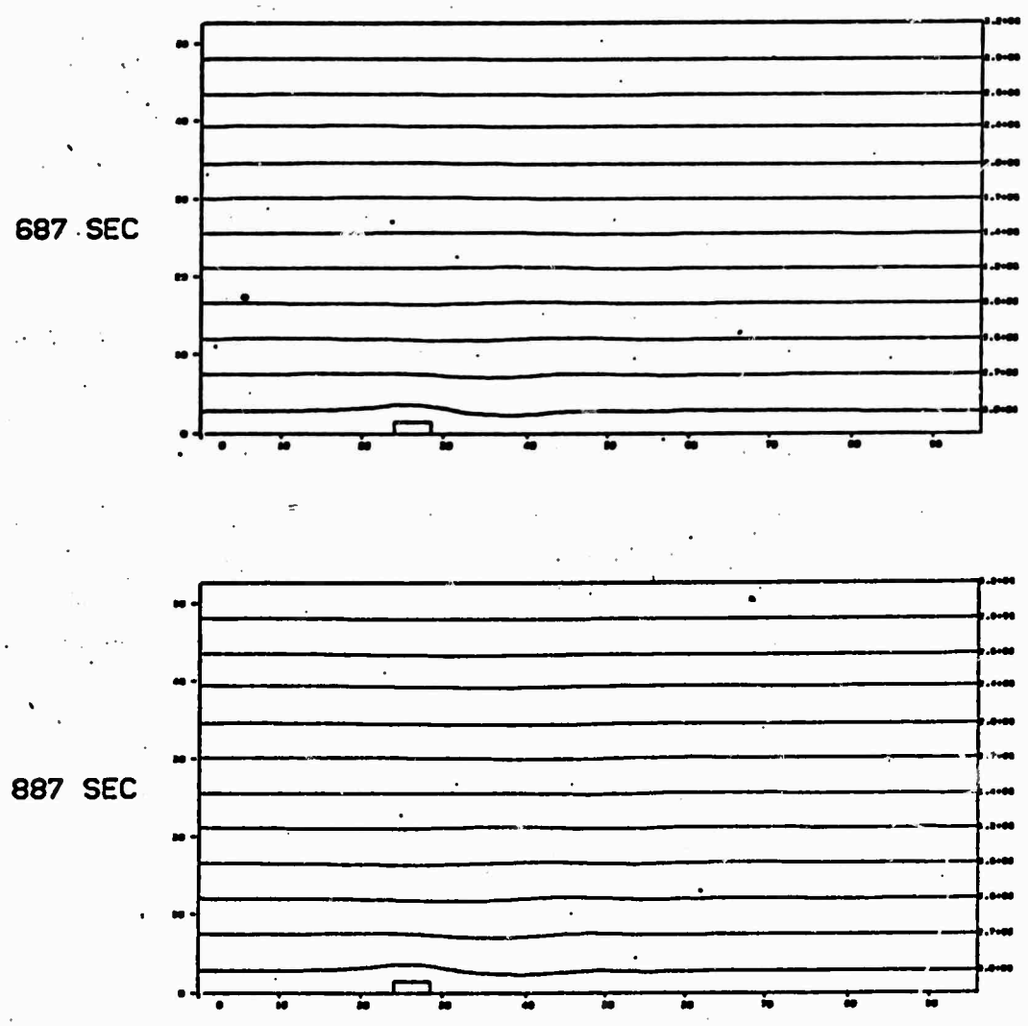


Figure 4.32 — Streamlines from Tropopause Problem (page 2).

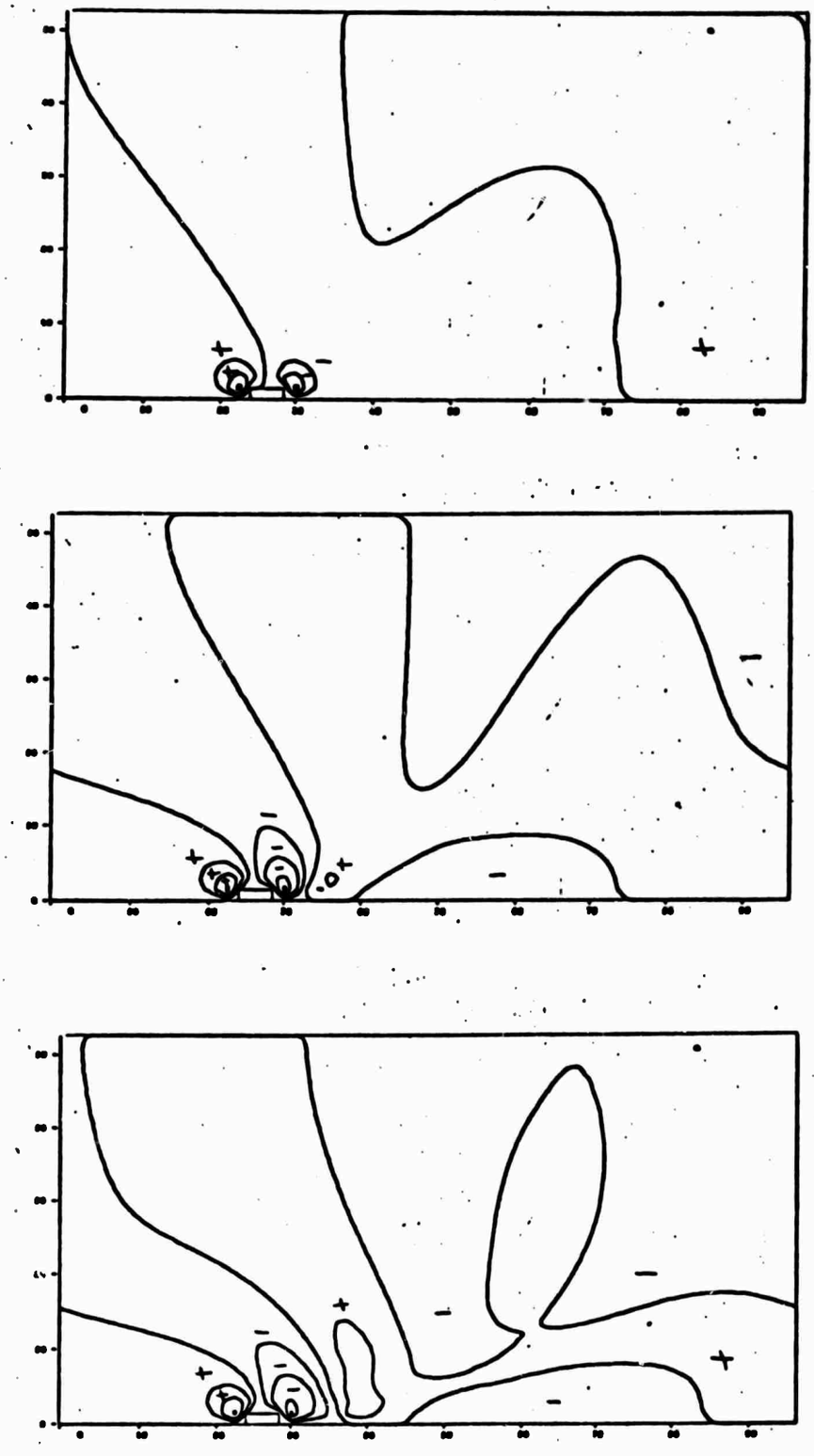


Figure 4.33 – Vertical Velocity Field from Tropopause Problem.

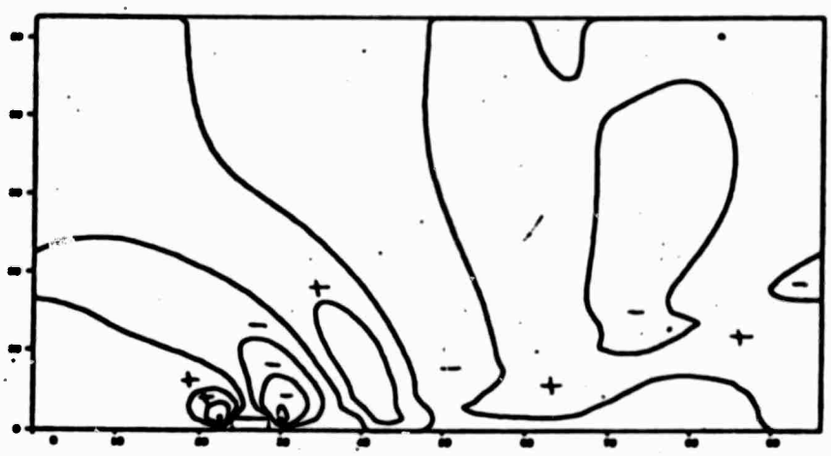
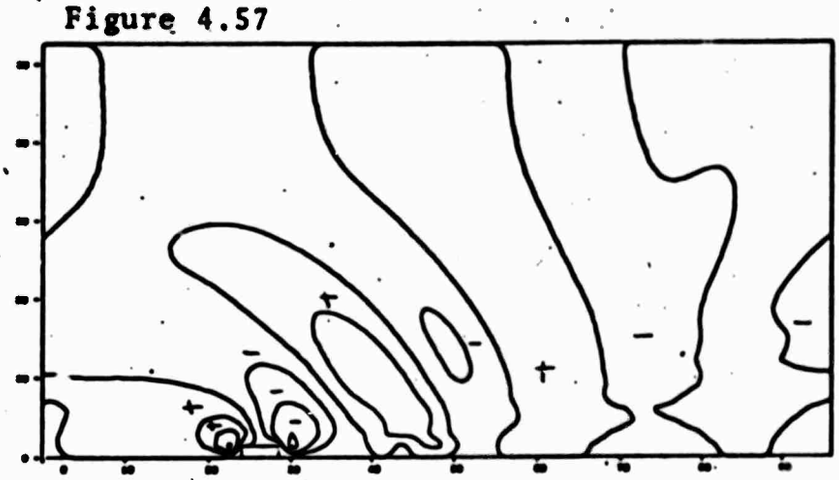


Figure 4.57



5. RADIATION IN THE EARTH'S ATMOSPHERE

The radiative transfer problem in the Earth's atmosphere reduces to the solution of the seemingly simple equation

$$\frac{dI_\nu}{ds} = J_\nu - \kappa_\nu I_\nu \qquad (5.1)$$

which states that radiant intensity, in traversing the element of length ds, will be augmented by sources in the amount $J_\nu\, ds$ and diminished by extinction in the amount $\kappa_\nu I_\nu ds$. In general, I_ν, the radiant intensity, and J_ν, the source function, depend on both a spatial coordinate $\vec{r}$ and an angular coordinate (direction) $\vec{\Omega}$ at the point $\vec{r}$, as well as upon the frequency ν. The time dependence of these quantities is ignored because the radiative state of the atmosphere is established, for all practical purposes, instantaneously. κ_ν is the extinction coefficient, which describes the relative depletion in the intensity of the beam, dI_ν/I_ν, upon traversing the element of distance ds. κ_ν is in general the sum of an absorption part and a scattering part. J_ν describes the additions made to the beam intensity along ds by thermal or non-thermal emission and by scattering. In the case where the source consists only of thermal emission, J_ν does not depend on I_ν and Eq. (5.1) can be solved explicitly for I_ν:

$$I_\nu(s,\vec{\Omega}) = I_\nu(s_o,\vec{\Omega}) \exp\left[-\int_{s_o}^{s} \kappa_\nu(s')\, ds'\right] + \int_{s_o}^{s} J_\nu(s'',\vec{\Omega}) \exp\left[-\int_{s''}^{s} \kappa_\nu(s')\, ds'\right] ds'' \qquad (5.2)$$

where s_o corresponds to some boundary at which I_ν is presumed known. However, even in the case where J_ν depends on I_ν, Eq. (5.2) is a perfectly valid alternate formulation of the radiative transfer equation. Eq. (5.1) will be called the differential form, and Eq. (5.2), the integral form, of the transfer equation.

The consideration will be limited to plane geometry, which is justified in view of the small depth of the Earth's atmosphere in comparison to its radius (when the sun is near the horizon the plane-parallel atmosphere approximation fails). The geometry is illustrated in Figure 5.1(a). The vertical coordinate is denoted by z, and the angular coordinates defining the direction $\vec{\Omega}$ at z are denoted by θ and ϕ. As is customary, the variable $\mu \equiv \cos\theta$ is employed rather than θ itself. From Figure 5.1(b), it is clear that in these coordinates the element of distance ds is related to dz by

$$ds = \frac{dz}{\mu} \quad ,$$

so that Eq. (5.1) becomes

$$\mu \frac{\partial I_\nu}{\partial z} = J_\nu - \kappa_\nu I_\nu \qquad (5.3)$$

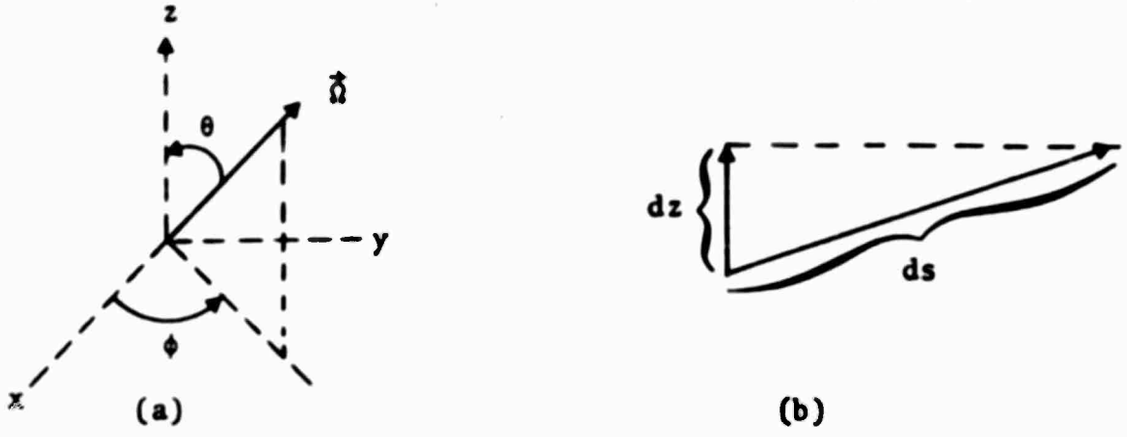


Figure 5.1 – Coordinate system for radiation problem.

where $I_\nu = I_\nu(z,\mu,\phi)$ and $\kappa_\nu = \kappa_\nu(z)$. If we make the assumption of local thermodynamic equilibrium (LTE), which is valid below about 70 km in the atmosphere, then the source term J_ν may be replaced by a more explicit expression, leading to the transfer equation

$$\mu \frac{\partial I_\nu}{\partial z} = \alpha'_\nu (B_\nu - I_\nu) + \beta_\nu \left[\frac{1}{4\pi} \int P_\nu(z,\vec{\Omega},\vec{\Omega}') \, I_\nu(z,\vec{\Omega}') \, d\Omega' - I_\nu \right] \quad (5.4)$$

where α'_ν is the volume absorption coefficient α_ν corrected for stimulated emission

$$\alpha'_\nu = \alpha_\nu \left(1 - e^{-h\nu/kT} \right) \quad , \quad (5.5)$$

β_ν is the scattering coefficient, B_ν is the Planck function

$$B_\nu(T) = \frac{2h\nu^3/c^2}{e^{h\nu/kT} - 1} \quad , \quad (5.6)$$

and P_ν is the phase function, defined so that

$$P_\nu(z,\vec{\Omega},\vec{\Omega}') \, \frac{d\Omega}{4\pi}$$

is the probability that a photon entering a volume element around z from direction $\vec{\Omega}'$ will be scattered into the cone $d\Omega$ of directions around $\vec{\Omega}$. Since absorption is explicitly represented in Eq. (5.4), the above probabilities must sum to one

$$\int_{4\pi} P_\nu(z,\Omega,\vec{\Omega}) \frac{d\Omega}{4\pi} = 1 \tag{5.7}$$

rather than to some number less than one as would be the case if absorption were tacitly included in the scattering terms.

Because a volume element in the atmosphere has no preferred direction with respect to scattering (due, say, to a permanent dipole moment), the scattering probability function P_ν depends only on the angle θ_s between $\vec{\Omega}$ and $\vec{\Omega}'$. If

$$\mu_s = \cos\theta_s$$

this means that $P_\nu = P_\nu(z,\mu_s)$. μ_s is expressed in terms of known angles θ, θ', ϕ, ϕ' as follows:

$$\begin{aligned}
\mu_s = \vec{\Omega}\cdot\vec{\Omega}' &= (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta) \\
&\cdot (\sin\theta'\cos\phi', \sin\theta'\sin\phi', \cos\theta') \\
&= \sin\theta\sin\theta'(\cos\phi\cos\phi' + \sin\phi\sin\phi') \\
&+ \cos\theta\cos\theta' \\
&= \mu\mu' + \sqrt{1-\mu^2}\sqrt{1-\mu'^2}\cos(\phi-\phi') \quad .
\end{aligned}$$

Let us integrate P_ν over all azimuthal directions ϕ ;

$$\int_0^{2\pi} P_\nu(z,\mu_s)d\phi = \int_0^{2\pi} P_\nu\left(z,\mu\mu' + \sqrt{1-\mu^2}\sqrt{1-\mu'^2}\cos(\phi-\phi')\right)d\phi$$

$$= \int_{-\phi'}^{2\pi-\phi'} P_\nu\left(z,\mu\mu' + \sqrt{1-\mu^2}\sqrt{1-\mu'^2}\cos\hat{\phi}\right)d\hat{\phi}$$

$$= \left[\int_0^{2\pi} + \int_{-\phi'}^{0} - \int_{2\pi-\phi'}^{2\pi}\right] P_\nu\left(z,\mu\mu' + \sqrt{1-\mu^2}\right.$$

$$\left. \cdot \sqrt{1-\mu'^2}\cos\hat{\phi}\right)d\hat{\phi} \quad .$$

The second and third integrals cancel one another, because of the periodicity of $\cos\hat{\phi}$, leaving

$$\bar{P}_\nu(z,\mu,\mu') \equiv \frac{1}{2\pi}\int_0^{2\pi} P_\nu(z,\mu_s)d\phi$$

$$= \frac{1}{2\pi}\int_0^{2\pi} P_\nu\left(z,\mu\mu' + \sqrt{1-\mu^2}\sqrt{1-\mu'^2}\cos\phi\right)d\phi \quad . \quad (5.8)$$

The important point to note here is that $\bar{P}_\nu$ does not depend on ϕ' .

Using Eq. (5.8), one may integrate Eq. (5.4) over azimuth ϕ and so deal only with the azimuthally-averaged intensity $\bar{I}_\nu$:

$$\mu \frac{\partial \bar{I}_\nu}{\partial z} = \alpha'_\nu (B_\nu - \bar{I}_\nu)$$

$$+ \beta_\nu \left[\frac{1}{2} \int_{-1}^{1} \bar{P}_\nu(z,\mu,\mu') \bar{I}_\nu(z,\mu') d\mu' - \bar{I}_\nu \right] \qquad (5.9)$$

where

$$\bar{I}_\nu(z,\mu) = \frac{1}{2\pi} \int_0^{2\pi} I_\nu(z,\mu,\phi) d\phi \quad .$$

In problems such as the solar aureole, the location of the neutral points in the sunlit sky, etc., it is clearly necessary to retain the ϕ-dependence of the intensity; for the computation of some important angular moments of the intensity, however, and in particular for the computation of vertical radiative fluxes and hence heating rates, $\bar{I}_\nu$ contains all the necessary information.

No more than the first three angular moments of the intensity will be considered in what follows. They are, in the customary notation,

$$E_\nu(z) = \frac{1}{c} \int I_\nu(z,\vec{\Omega}) d\Omega = \frac{2\pi}{c} \int_{-1}^{1} \bar{I}_\nu(z,\mu) d\mu$$

$$F_{\alpha,\nu}(z) = \int \Omega_\alpha \, I_\nu(z,\vec{\Omega}) d\Omega \qquad (5.10)$$

$$P_{\alpha\beta,\nu}(z) = \frac{1}{c} \int \Omega_\alpha \, \Omega_\beta \, I_\nu(z,\vec{\Omega}) d\Omega$$

where

$$\Omega_x = \sqrt{1-\mu^2}\cos\phi \ , \qquad \Omega_y = \sqrt{1-\mu^2}\sin\phi \ , \qquad \Omega_z = \mu \ .$$

E , F , and P may be interpreted physically as the density of radiant energy, flux of radiant energy, and radiation pressure tensor, respectively. Two moments of special interest are the vertical flux F_z and the zz-component of pressure P_{zz} :

$$F_\nu(z) = \int \mu \ I_\nu(z,\vec{\Omega})\,d\Omega$$

$$= 2\pi \int_{-1}^{1} \mu \ \bar{I}_\nu(z,\mu)\,d\mu \qquad (5.11)$$

$$P_\nu(z) = \frac{1}{c} \int \mu^2 I_\nu(z,\vec{\Omega})\,d\Omega$$

$$= \frac{2\pi}{c} \int_{-1}^{1} \mu^2 \bar{I}_\nu(z,\mu)\,d\mu$$

where for conveience the coordinate subscripts have been omitted. The other components of the flux, F_x and F_y , might be of interest for some applications, such as the heating of inclined slopes, but their calculation would require retaining the ϕ-dependence of I_ν , as would the calculation of any of the other pressure components.

In general, radiation energies and pressures within the atmosphere are completely negligible compared to material energies and pressures, whereas radiative fluxes are comparable to other energy fluxes in the atmosphere (latent heat, sensible heat, etc.); the reason for this is that the relatively small amounts of radiant energy travel at the speed of light, while the speed of material energy propagation is essentially limited by the sound speed.

The definitions of radiation energy and flux given above are made plausible by looking at the result of integrating Eq. (5.9) over all μ (remembering the normalization Eq. (5.7) of P_ν):

$$\frac{dF_\nu}{dz} = \alpha'_\nu(4\pi B_\nu - cE_\nu) \quad . \tag{5.12}$$

The terms on the right-hand side are source and sink terms to the radiation energy field; if they cancel, then Eq. (5.12), with the above interpretation of F_ν, becomes the usual expression for steady-state radiative energy conservation. If they do not cancel, then clearly more energy is entering an infinitesimal horizontal layer than is leaving it, or vice versa, and the deposited or withdrawn energy will result in a net heating or cooling of that layer. The expression for the heating rate is, in fact, taking the origin of z-coordinates at the surface of the Earth

$$\rho C_p \frac{dT}{dt} = - \frac{dF}{dz} \quad . \tag{5.13}$$

where

$$F(z) = \int_0^\infty F_\nu(z)\,d\nu \qquad (5.14)$$

$\rho(z)$ = density of air at z

C_p = specific heat of air at constant pressure .

Eq. (5.13) is simply a restatement of the first law of thermodynamics, which is, in the usual notation,

$$dq = de + p\,dv$$

$$= C_v\,dT + \rho RT\,d\left(\frac{1}{\rho}\right)$$

$$= C_v\,dT - \frac{RT}{\rho}\,d\rho$$

for an ideal gas. Since any atmospheric process at a given level z will for all practical purposes take place at constant pressure,

$$dp = 0 = R(\rho\,dT + T\,d\rho)$$

then the first law can be written

$$dq = C_v\,dT - \frac{R}{\rho}(-\rho\,dT)$$

$$= (C_v + R)\,dT = C_p\,dT \quad .$$

Dividing by dt , and noting that the heating rate due to radiation $\frac{dF}{dz}$ corresponds to $\frac{d(\rho q)}{dt}$, leads to Eq. (5.13) (the factor ρ converts heating per unit mass to heating per unit volume, in agreement with the definition of F).

To obtain the radiative heating rates in Eq. (5.13) is the primary goal of our calculation. To arrive at these numbers, Eq. (5.9) (or its corresponding integral form) must be solved for the $\bar{I}_\nu$'s , which must be integrated according to Eq. (5.11) to obtain the F_ν's , and finally the F_ν's must be integrated in accordance with Eq. (5.14) to obtain the F's . The mechanics of solving Eq. (5.9) will be elaborated upon in the remainder of this section.

The solution of Eq. (5.9) can be separated into several sub-tasks, which are:

(1) specification of the phenomenological parameters entering the equation;

(2) specification of boundary conditions; and

(3) discretization of the independent variables μ , z , ν for numerical solution.

These three sub-tasks are discussed in turn in Sections 5.1 through 5.3.

5.1 PARAMETER SPECIFICATION

The parameters required in Eq. (5.9) are the absorption coefficient α'_ν , the scattering coefficient β_ν , and the ϕ-averaged phase function $\bar{P}_\nu$. α'_ν and β_ν are known to have both a temperature (T) and pressure (p) dependence, so that the temperature and pressure structure of the atmosphere constitutes required input data (perhaps from a GCM). Other pertinent input data, required for the computation of β_ν and $\bar{P}_\nu$, are the aerosol and cloud structure of the atmosphere; more specifically, the number density of aerosol particles (including cloud droplets) as a function of both

height and particle radius, and the frequency dependent index of refraction of all aerosol constituents. For the computation of optical path lengths, the mixing ratios of the non-uniformily mixed gases H_2O and O_3 as a function of height are also required.

Needless to say, the specification of atmospheric structure in such detail is beyond the capabilities of either experiment (soundings) or theory (GCM's) at the present time. However, experimentation with the detailed model being developed should point the way toward simpler specifications of structure which are nevertheless sufficient for computing heating rates. At the same time, one may expect an improvement in vertical resolution and aerosol prediction capability in the GCM's and more accurate experimental data, particularly with regard to aerosols, in the near future. Hopefully, this will lead to a felicitous convergence of the radiation model's need for atmospheric structure data and the ability of theory and/or experimental to furnish it.

The absorption coefficient α_ν will not be discussed in detail here. Absorption in the atmosphere takes place in large part in vibration-rotation bands of H_2O, CO_2, O_3 and other minor constituents; each band contains thousands of spectral lines, resulting in an extremely rapid variation of α_ν with frequency. Voluminous compilations of α_ν exist, but it is impossible, for reasons of computer storage and economy, to discretize $\bar{I}_\nu$ in frequency space finely enough to follow the variations of α_ν; spectral intervals must instead be taken which contain many lines. Hence, this whole discussion falls more naturally into the province of Section 5.3, where discretization in ν-space is discussed.

The Rayleigh scattering coefficient $\beta_{\nu,R}$ and phase function P_R (independent of ν) are discussed by Penndorf.[15]

He gives the index of refraction n_s of air at $p = 760$ mm Hg, $T = 15^0C$, and water vapor pressure f (in mm Hg) as

$$(n_s - 1) \times 10^{-6} = 64.328 + 29498.1\left(146 - \frac{1}{\lambda^2}\right)^{-1} + 255.4\left(41 - \frac{1}{\lambda^2}\right)^{-1} - \left(0.0624 - \frac{0.00068}{\lambda^2}\right)\frac{f}{1.0549}$$

where λ is the vacuum wavelength in microns. The Rayleigh volume scattering coefficient is then

$$\beta_{\nu,R} = \frac{8\pi^3}{3}\frac{(n_s^2 - 1)^2}{\lambda^4 N_s^2}\left(\frac{6 + 3\rho_n}{6 - 7\rho_n}\right)N \qquad (5.15)$$

and the Rayleigh phase function[16] is

$$P_R(\mu_s) = \frac{3(1+\rho_n)}{4 + 2\rho_n}\left(1 + \frac{1-\rho_n}{1+\rho_n}\mu_s^2\right) \qquad (5.16)$$

where $\mu_s = \cos\theta_s$ = cosine of scattering angle

N_s = number density of air molecules at 760 mm Hg and $15^0C = 2.54743 \times 10^{19}$ cm^{-3}

ρ_n = depolarization factor

N = number density of air molecules at p and T of interest.

The factor involving ρ_n expresses the effect of the optically anisotropic molecules upon the scattering, and its value has

been calculated and measured by a number of investigators since the effect was first discovered (see Penndorf, Table II). Penndorf chooses $\rho_n = 0.035$, which we shall use in our work. The number density N may be evaluated from the perfect gas law as

$$N = \frac{p}{kT}$$

where k is Boltzmann's constant, T is in ^0K, and p is in compatible units.

The scattering from aerosols in the atmosphere (including clouds) may be treated by the Mie theory. This involves a certain degree of approximation, in that the Mie theory assumes spherical particles and natural atmospheric aerosols may not be spherical (although water droplets, the most important aerosols, are indeed spherical provided they do not have an appreciable fall velocity). Also, the complex indices of refraction

$$m = n_1 - i\, n_2 \tag{5.17}$$

of the aerosols, which are required by the Mie theory, are in many cases not well-known, especially in the infrared. Nevertheless, Mie theory, despite its well-known mathematical and computational complexities, is the only reasonable approach to aerosol scattering currently available.

If a is the radius of a single spherical particle, having index of refraction m relative to the surrounding medium, then the scattering pattern of that particle for light of wavelength λ can be described in terms of the following two functions:[16]

$$i_1 = \left| \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} \left[a_n(\alpha,m)\pi_n(\mu_s) + b_n(\alpha,m)\tau_n(\mu_s) \right] \right|^2 \qquad (5.18)$$

$$i_2 = \left| \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} \left[a_n(\alpha,m)\tau_n(\mu_s) + b_n(\alpha,m)\pi_n(\mu_s) \right] \right|^2 \qquad (5.19)$$

where

$$\alpha = \frac{2\pi a}{\lambda} \qquad (5.20)$$

$$\beta = m\alpha \qquad (5.21)$$

$$\mu_s = \cos\theta_s$$

$$a_n = \frac{\psi_n(\alpha)\psi_n'(\beta) - m\,\psi_n(\beta)\psi_n'(\alpha)}{\zeta_n(\alpha)\psi_n'(\beta) - m\,\psi_n(\beta)\zeta_n'(\alpha)} \qquad (5.22)$$

$$b_n = \frac{\psi_n(\beta)\psi_n'(\alpha) - m\,\psi_n(\alpha)\psi_n'(\beta)}{\psi_n(\beta)\zeta_n'(\alpha) - m\,\zeta_n(\alpha)\psi_n'(\beta)} \qquad (5.23)$$

$$\pi_n(\mu) = P_n'(\mu) \qquad (5.24)$$

$$\tau_n(\mu) = \mu\pi_n(\mu) - (1-\mu^2)\pi_n'(\mu) \quad . \qquad (5.25)$$

The P_n are Legendre polynomials. The ψ_n, ζ_n are called Ricatti-Bessel functions, and are defined in terms of the more familiar spherical Bessel functions j_n, y_n, etc. by

$$\psi_n(z) = zj_n(z)$$

$$\chi_n(z) = -zy_n(z)$$

$$\zeta_n(z) = \psi_n + i\chi_n = zh_n^{(2)}(z) \quad .$$

For unpolarized incident light, the distribution of scattered intensity from the spherical particle is proportional to $i_1 + i_2$. An unpolarized incident beam is also tacitly assumed in the form (5.16) of the Rayleigh scattering pattern. The full polarization-dependent treatment of radiative transfer, in which the intensity is replaced by a four-component vector and all the phase functions are replaced by 4×4 phase matrices, involves a great deal more computing than the present method for a relatively small improvement in accuracy[17] (the largest reported errors in I_ν from neglecting polarization are about 10 percent, with more typical values being 1-5 percent). Therefore, the present model will be constructed assuming every Mie scattering event produces only unpolarized light, of intensity proportional to $i_1 + i_2$.

The extinction, scattering, and absorption cross-sections for the spherical particle may be computed to be

$$\sigma_{ext} = \frac{\lambda^2}{2\pi} \sum_{n=1}^{\infty} (2n+1)\ \mathrm{Re}(a_n+b_n)$$

$$\sigma_{sca} = \frac{\lambda^2}{2\pi} \sum_{n=1}^{\infty} (2n+1)(|a_n|^2 + |b_n|^2)$$

$$\sigma_{abs} = \sigma_{ext} - \sigma_{sca}$$

Presuming that the atmospheric aerosol at a given height contains a number density N_{aer} of spherical particles, with a probability distribution $n(a)$ of radii such that

$$n(a)da = \text{fraction of particles with radii in } (a, a+da)$$

and

$$\int_{a_{min}}^{a_{max}} n(a)da = 1$$

then it can be shown that the volume scattering and absorption coefficients for this aerosol are

$$\beta_{\nu,M} = N_{aer} \int_{a_{min}}^{a_{max}} \sigma_{sca}(\alpha)\, n(a)da \tag{5.26}$$

and

$$\alpha_{\nu,M} = N_{aer} \int_{a_{min}}^{a_{max}} \sigma_{abs}(\alpha)\, n(a)da \tag{5.27}$$

respectively. The phase function will be

$$P_{\nu,M} = 4\pi \frac{\int_{a_{min}}^{a_{max}} (i_1 + i_2)\, n(a)da}{\int d\Omega \int_{a_{min}}^{a_{max}} (i_1 + i_2)\, n(a)da}$$

where the integral over $d\Omega = d\mu_s d\phi_s$ is an integral over all scattering angles; its appearance in the denominator guarantees the proper normalization of $P_{\nu,M}$ to 4π. From the expression for σ_{sca} in terms of i_1 and i_2,

$$\sigma_{sca} = \frac{\lambda^2}{2\pi} \int_{-1}^{1} \left[i_1(\mu_s) + i_2(\mu_s) \right] d\mu_s$$

and from Eq. (5.26), the phase function may be written

$$P_{\nu,M}(\mu_s) = \frac{\lambda^2 N_{aer}}{\pi \beta_{\nu,M}} \int_{a_{min}}^{a_{max}} \left[i_1(\mu_s) + i_2(\mu_s) \right] n(a) da \quad . \qquad (5.28)$$

P_R and $P_{\nu,M}$ may be combined as follows to yield the complete phase function for scattering:

$$P_\nu = \frac{\beta_{\nu,R} P_R + \beta_{\nu,M} P_{\nu,M}}{\beta_\nu} \qquad (5.29)$$

where

$$\beta_\nu = \beta_{\nu,R} + \beta_{\nu,M} \qquad (5.30)$$

is the total volume scattering coefficient.

Details as to the actual computation of i_1, i_2, $\beta_{\nu,M}$, and $P_{\nu,M}$ are omitted here for brevity. The Mie program has been coded and debugged using existing tables of Mie functions, and every effort has been expended to keep its cost minimal, in view of the fact that it is only one small part of the radiation program. An elaboration of the numerical techniques used will be furnished in the final report.

5.2 BOUNDARY CONDITIONS

As is clear from, for example, Figure 1.1 of Goody,[18] the spectrum of solar radiation and the spectrum of terrestrial radiation overlap hardly at all. Therefore, we will speak of the atmospheric radiation problem as two separate problems, and discuss the boundary conditions for each problem separately.

5.2.1 Solar Spectrum

The following data completely specify the boundary conditions for the solar radiation problem:

- solar zenith angle
- solar constant (flux of solar energy at the top of the atmosphere)
- solar spectral energy distribution
- albedo or reflection coefficient at the surface of the Earth.

The solar zenith angle at any particular location on the Earth's surface is a function of the day of the year and the time of day. Since time zones are so irregular as to be virtually meaningless, all times will be taken as Greenwich mean times. Then the solar zenith angle $\theta_{sun} \varepsilon [0^0, 90^0]$ will be computed from the four variables latitude, longitude, calendar date, and Greenwich mean time. Leap years will be accounted for. By having the capability to specify the solar zenith angle in this fashion, comparisons can be made between the model's predictions and experimental data gathered at any time in the past.

The solar constant is still, surprisingly, a subject of debate. The best values of both the solar constant and the solar spectral energy distribution seem to be those of

Thekaekara and his collaborators[19] which will be used in the present code. The solar constant will be adjusted according to calendar date because of the varying Earth-sun distance, which can alter the solar constant ±3½ percent from its mean value. Taking the origin of vertical coordinates $z = 0$ at the surface of the Earth, the solar boundary condition can be expressed mathematically as

$$I_\nu(z_o,\mu,\phi) = S_\nu \, \delta(\vec{\Omega} - \vec{\Omega}_{sun})$$

for

$$-1 \leq \mu \leq 0 \qquad\qquad 0 \leq \phi \leq 2\pi$$

where z_o is the vertical coordinate of the "top" of the atmosphere and S_ν is the solar intensity at frequency ν. In terms of the azimuthally-averaged intensity, this is

$$\bar{I}_\nu(z_o,\mu) = \frac{S_\nu}{2\pi} \delta(\mu - \mu_{sun}) \qquad\qquad -1 \leq \mu \leq 0$$

where

$$\mu_{sun} = -\cos \theta_{sun} \quad .$$

The sun is, of course, not a δ-function but has a finite angular width, of about ½°. A more realistic boundary condition was used in the heating rate equation for no scattering to estimate the error of using the δ-function. The conclusion was that the absolute error produced in the heating rate is always inconsequential (whereas the relative error in the heating rate could become substantial near sunrise or sunset, $\mu \gtrsim 0$) .

The albedo of a surface is the ratio of the outgoing to the incoming flux. Since intensities, not fluxes, are being calculated, the specification of the albedo alone is not sufficient. The distribution of the outgoing or reflected intensity in angle is also required. The assumption is often made that the Earth's surface is a "Lambert reflector," meaning that the reflected intensity is isotropic and unpolarized regardless of the angular distribution and polarization of the incident radiation. Rough, irregular surfaces approximate Lambert reflectors. If the incident flux, computed from the incident intensities, is $F_{\nu,inc}$, and the albedo (which may be frequency-dependent) is A_ν , then for a Lambert or diffuse reflector

$$F_{\nu,ref} = 2\pi \int_0^1 \mu \bar{I}_\nu(0,\mu)d\mu = \pi \bar{I}_\nu(0,\mu) = A_\nu \left| F_{\nu,inc} \right| \quad .$$

Thus, the reflected intensities $\bar{I}_\nu(0,\mu)$, $0 \leq \mu \leq 1$, are specified in terms of the albedo A_ν and an integral $F_{\nu,inc}$ of the incident intensities:

$$\bar{I}_\nu(0,\mu) = 2A_\nu \left| \int_{-1}^0 \mu \bar{I}_\nu(0,\mu)d\mu \right| \qquad 0 \leq \mu \leq 1 \quad .$$

Measurements of albedo indicate a more complicated situation than that described above. On cloudless days, the albedo seems to be fairly constant for solar zenith angles $\theta_{sun} \leq 60^0$, and to increase rather markedly as θ_{sun} increases from 60^0 to 90^0. In particular, this phenomenon is observed for the ocean and for desert and semi-desert areas.[20] Therefore, let us define a directional albedo, or directional-hemispherical reflectivity, $A_\nu(\mu)$, which is the reflected

flux divided by (and due to) an incident flux from the direction (θ,ϕ). ($\theta \in [0^\circ, 90^\circ]$ is the angle to the surface normal.) Then if the incident intensity from direction $\vec{\Omega}$ is $I_\nu(0,\mu,\phi)$, $-1 \leq \mu \leq 0$, the incident flux will be

$$|\mu| \; I_\nu(0,\mu,\phi) \; d\Omega$$

which will cause a reflected flux

$$A_\nu(|\mu|) \; |\mu| \; I_\nu(0,\mu,\phi) \; d\Omega \quad .$$

Summing over all incident directions leads to the total reflected flux

$$F_{\nu,ref} = \int_0^{2\pi} d\phi \int_{-1}^{0} d\mu \; |\mu| \; A_\nu(|\mu|) \; I_\nu(0,\mu,\phi)$$

$$= 2\pi \int_{-1}^{0} |\mu| \; A_\nu(|\mu|) \; \bar{I}_\nu(0,\mu) \; d\mu \quad .$$

For a diffuse reflector, this implies a reflected intensity of

$$\bar{I}_\nu(0,\mu) = 2 \int_{-1}^{0} |\mu| \; A_\nu(|\mu|) \; \bar{I}_\nu(0,\mu) \; d\mu \quad , \qquad (5.31)$$

$$0 \leq \mu \leq 1 \quad .$$

Clearly, the directional albedo $A_\nu(\mu)$ contains no information as to the angular distribution of the reflected radiation. Such information is furnished in complete detail by the bidirectional reflectivity ρ_ν ,[21] which is equal to

the reflected intensity $I_{\nu,ref}(\theta,\phi,\theta_r,\phi_r)$ at angles θ_r, ϕ_r, due to an incident intensity $I_{\nu,inc}(\theta,\phi)$ at angles θ, ϕ, divided by the flux of that incident intensity:

$$\rho_\nu(\theta,\phi,\theta_r,\phi_r) = \frac{\pi I_{\nu,ref}(\theta,\phi,\theta_r,\phi_r)}{I_{\nu,inc}(\theta,\phi)\cos\theta\, d\Omega} .$$

The various angles are defined in Figure 5.2. If the reflecting surface is isotropic, as the Earth's surface largely is, ρ_ν will depend only on the difference $\phi - \phi_r$, $\rho_\nu = \rho_\nu(\theta,\theta_r,\phi-\phi_r)$. Since $I_{\nu,ref}$ is of differential order with respect to $I_{\nu,inc}$ (except in the case of specular reflection), the $d\Omega$ in the denominator keeps ρ_ν from being of differential order. The factor π is introduced so that, if the reflection is diffuse ($I_{\nu,ref}$ independent of θ_r, ϕ_r), ρ_ν reduces to the directional albedo A_ν defined above.

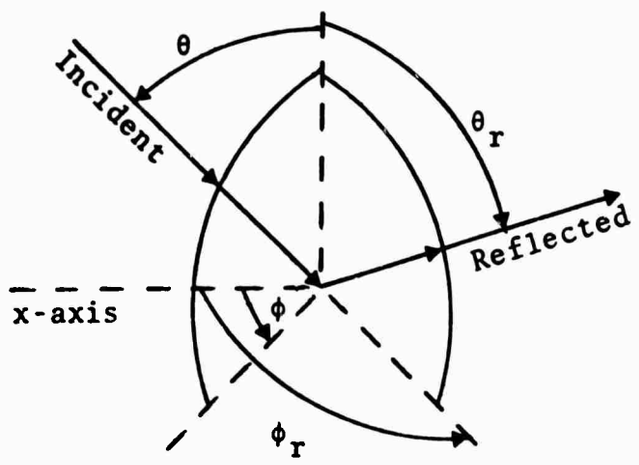


Figure 5.2 – Geometry of reflection.

The total reflected intensity at angles θ_r, ϕ_r is found by summing $I_{\nu,ref}$ over all possible angles θ, ϕ of incidence:

$$I_{\nu,ref}(\theta_r,\phi_r) = \sum_{\substack{\text{incident}\\ \text{angles}}} I_{\nu,ref}(\theta,\phi,\theta_r,\phi_r)$$

$$= \frac{1}{\pi}\int_0^{2\pi} d\phi \int_0^{\pi/2} d\theta \sin\theta\, \rho_\nu(\theta,\theta_r,\phi-\phi_r)$$

$$\cdot I_{\nu,inc}(\theta,\phi)\cos\theta \ .$$

Phrasing this in our usual notation,

$$I_\nu(0,\mu,\phi) = \frac{1}{\pi}\int_0^{2\pi} d\phi' \int_{-1}^{0} d\mu'|\mu'|\, \rho_\nu(|\mu'|,\mu,\phi'-\phi)$$

$$\cdot I_\nu(0,\mu',\phi') \quad \text{for} \quad 0 \le \mu \le 1 \ . \tag{5.32}$$

Because ρ_ν must be periodic in its third argument,

$$\rho_\nu(\mu,\mu',\phi+2\pi) = \rho_\nu(\mu,\mu',\phi)$$

the azimuthally-averaged bidirectional reflectivity

$$\overline{\rho_\nu} = \frac{1}{2\pi}\int_0^{2\pi} \rho_\nu(\mu,\mu',\phi-\phi')\, d\phi$$

can be reduced to

$$\overline{\rho_\nu}(\mu,\mu') = \frac{1}{2\pi}\int_0^{2\pi} \rho_\nu(\mu,\mu',\hat{\phi})\, d\hat{\phi}$$

which is independent of ϕ' . This result makes it possible to azimuthally average Eq. (5.32) to yield

$$\bar{I}_\nu(0,\mu) = 2 \int_{-1}^{0} d\mu' \, |\mu'| \, \overline{\rho_\nu}(|\mu'|,\mu) \, \bar{I}_\nu(0,\mu') \ , \qquad (5.33)$$

$$0 \leq \mu \leq 1 \ .$$

This is the most general reflective boundary condition.

For water surfaces, which cover about three-fourths of the Earth, computations of ρ_ν are possible in terms of the Fresnel formulas for reflection, the index of refraction $m = n_1 - i\, n_2$ of water, and a statistical distribution of surface slopes (as a function of wind speed). The work of Chow[22] is exemplary in this regard, although he ignores the imaginary part n_2 of the index of refraction in the IR and uses a frequency-averaged value of n_1 in his computations. He also uses the simplest analytical approximation to Cox and Munk's[23] experimentally-determined sea-slope distributions. We have replaced these approximations with more accurate ones and computed tables of $\overline{\rho_\nu}$ for use in Eq. (5.33). The tabulation is somewhat simplified by the reciprocity relation for ρ_ν[21]

$$\rho_\nu(\theta,\theta_r,\phi-\phi_r) = \rho_\nu(\theta_r,\theta,\phi_r-\phi)$$

which implies

$$\overline{\rho_\nu}(\mu,\mu') = \overline{\rho_\nu}(\mu',\mu) \ .$$

It should be noted that only surface reflection is accounted for in the above computation of ρ_ν . Backscattering

from turbidity (primarily micro-organisms) beneath the surface is not accounted for, although measurements in the Russian literature indicate that this effect is only important at low solar elevations.[24]

The situation vis-á-vis reflectivity data for land surfaces is much worse than for the sea surface (cf. Kondrat´yev, Ref. 24). In general, only albedos are available, and often not even as a function of frequency. Therefore, the code will have several options. All options will use Eq. (5.33); however, if only directional albedos $A_\nu(\mu)$ are available, diffuse reflection will be assumed so that Eq. (5.33) reduces to Eq. (5.31). If only albedos A_ν are available, it will be presumed that $A_\nu(\mu) = A_\nu$, independent of μ. And, if only frequency-averaged albedos A are available, the code will take $A_\nu \equiv A$. Thus, as improved albedo data become available, they need only be entered into the appropriate tables and certain option flags re-set.

At this point it is convenient to introduce an additive splitting of the downward-directed intensity:

$$\bar{I}_\nu = \bar{I}_\nu^{\text{solar}} + \bar{I}_\nu^{\text{diff}} \quad , \quad -1 \leq \mu \leq 0 \quad . \tag{5.34}$$

$\bar{I}_\nu^{\text{solar}}$ is the solar beam intensity, and $\bar{I}_\nu^{\text{diff}}$ is the "diffuse" intensity produced by scattering and reflection (thermal emission being practically negligible in the solar spectrum). The reason for introducing this splitting is that the solar part of the intensity is essentially a δ-function in angle and so exceedingly difficult to represent numerically. The remaining part of the intensity, $\bar{I}_\nu^{\text{diff}}$, is usually smoothly behaved as a function of angle and so will not require nearly as fine an angular mesh for its representation as the original intensity $\bar{I}_\nu$ would have.

The solar beam intensity can be found from the transport equation in which only extinction processes are included:

$$\mu \frac{\partial \bar{I}_\nu^{solar}}{\partial z} = -\kappa_\nu(z) \bar{I}_\nu^{solar} \quad , \qquad -1 \leq \mu \leq 0$$

where $\kappa_\nu = \alpha'_\nu + \beta_\nu$. Imposing the boundary condition

$$\bar{I}_\nu^{solar}(z_o,\mu) = \frac{S_\nu}{2\pi} \delta(\mu-\mu_{sun}) \quad , \qquad -1 \leq \mu \leq 0$$

leads to the solution

$$\bar{I}_\nu^{solar} = \frac{S_\nu}{2\pi} \delta(\mu-\mu_{sun}) \exp\left[\frac{1}{\mu} \int_z^{z_o} \kappa_\nu(z')\, dz'\right] ,$$

$$-1 \leq \mu \leq 0 \quad .$$

Introducing the splitting (5.34), with the above solution for $\bar{I}_\nu^{solar}$, into the full transport equation, (5.9), one finds

$$\begin{aligned} \mu \frac{\partial \bar{I}_\nu^{diff}}{\partial z} = {} & \alpha'_\nu (B_\nu - \bar{I}_\nu^{diff}) \\ & + \beta_\nu \left[\frac{1}{2} \int_{-1}^{0} \bar{P}_\nu(z,\mu,\mu')\, \bar{I}_\nu^{diff}(z,\mu')\, d\mu' \right. \\ & \left. + \frac{1}{2} \int_0^1 \bar{P}_\nu(z,\mu,\mu')\, \bar{I}_\nu(z,\mu')\, d\mu' - \bar{I}_\nu^{diff}\right] \\ & + \frac{\beta_\nu S_\nu}{4\pi} \bar{P}_\nu(z,\mu,\mu_{sun}) \exp\left[\frac{1}{\mu_{sun}} \int_z^{z_o} \kappa_\nu(z')\, dz'\right] \end{aligned} \qquad (5.35)$$

for $-1 \leq \mu \leq 0$, and

$$\mu \frac{\partial \bar{I}_\nu}{\partial z} = \alpha'_\nu (B_\nu - \bar{I}_\nu)$$

$$+ \beta_\nu \left[\frac{1}{2} \int_{-1}^{0} \bar{P}_\nu(z,\mu,\mu') \, \bar{I}_\nu^{diff}(z,\mu') \, d\mu' \right.$$

$$\left. + \frac{1}{2} \int_{0}^{1} \bar{P}_\nu(z,\mu,\mu') \, \bar{I}_\nu(z,\mu') \, d\mu' - \bar{I}_\nu \right] \qquad (5.36)$$

$$+ \frac{\beta_\nu S_\nu}{4\pi} \bar{P}_\nu(z,\mu,\mu_{sun}) \exp\left[\frac{1}{\mu_{sun}} \int_{z}^{z_0} \kappa_\nu(z') \, dz' \right]$$

for $0 \leq \mu \leq 1$.

The boundary condition on $\bar{I}_\nu^{diff}$ at the top of the atmosphere is now simply a homogeneous one,

$$\bar{I}_\nu^{diff}(z_o,\mu) = 0 \qquad , \qquad -1 \leq \mu \leq 0 \quad . \qquad (5.37)$$

The reflective boundary condition at the surface, Eq. (5.33), becomes

$$\bar{I}_\nu(0,\mu) = 2 \int_{-1}^{0} |\mu'| \, \bar{\rho}_\nu(\mu,|\mu'|) \, \bar{I}_\nu^{diff}(0,\mu') \, d\mu'$$

$$+ \frac{S_\nu}{\pi} |\mu_{sun}| \, \bar{\rho}_\nu(\mu,|\mu_{sun}|) \exp\left[\frac{1}{\mu_{sun}} \int_{0}^{z_o} \kappa_\nu(z') \, dz' \right]$$

$$0 \leq \mu \leq 1 \quad . \qquad (5.38)$$

The extra terms in Eqs. (5.35) and (5.36) are interpretable physically as scattering sources due to the solar beam; the extra term in Eq. (5.38) is attributable to reflection of the solar beam.

A final boundary quantity, one of paramount interest, is the solar radiative flux into the ground, which determines the heating. It is

$$F_\nu(0) = 2\pi \int_{-1}^{1} \mu \bar{I}_\nu(0,\mu)\, d\mu$$

$$= 2\pi \int_{-1}^{0} \mu \bar{I}_\nu^{diff}(0,\mu)\, d\mu + 2\pi \int_{0}^{1} \mu \bar{I}_\nu(0,\mu)\, d\mu \qquad (5.39)$$

$$+ \mu_{sun}\, S_\nu \exp\left[\frac{1}{\mu_{sun}} \int_0^{z_o} \kappa_\nu(z')\, dz'\right] .$$

When this is integrated over frequency and added to the total terrestrial radiative flux out of the ground, the resultant flux determines a boundary condition for a ground heating calculation. If the flux is entirely absorbed within the first millimeter or so, as is the case for most land surfaces, then it determines a surface heat source; if it penetrates, as in the oceans and ice, a distributed heat source is determined. A code which solves the heat condition equation with prescribed sources has been developed at S^3 during this contract period, and will be coupled to the radiation code for studies of the dynamic interaction of the radiation field and the surface. It seems likely that the development of this coupled radiation-surface code will be only partially completed by the end of the present contract period, and will require further work in the follow-on period.

5.2.2 Terrestrial Radiation Spectrum

The relevant data required to specify the terrestrial radiation boundary condition are:

- o the surface temperature, T_g
- o the surface emissivity ε_ν, possibly as a function of the angle (to the surface normal) θ of emission.

The surface temperature T_g is presumed either given as an input variable or calculated by the ground heating code discussed in Section 5.2.1.

The directional emissivity is defined as the ratio of the thermally emitted intensity $I_\nu(\theta,\phi)$ in a particular direction to the black body intensity:

$$\varepsilon_\nu(\theta,\phi) = \frac{I_\nu(\theta,\phi)}{B_\nu(T_g)} \quad . \tag{5.40}$$

We shall only consider isotropic surfaces, so that the dependence on azimuthal angle ϕ disappears, $\varepsilon_\nu = \varepsilon_\nu(\theta)$ or $\varepsilon_\nu = \varepsilon_\nu(\mu)$. If the emitted intensity I_ν is independent of θ, ε_ν reduces to the more familiar hemispherical emissivity (ratio of emitted flux to black body flux $\pi B_\nu(T_g)$). The hemispherical and directional emissivities have been measured for many kinds of surfaces.[25] The angular behavior of ε_ν is similar for all electrical non-conductors (in particular, the Earth's surface); it is nearly constant and close to one for θ between 0^0 and 60^0, then it falls off to zero as θ increases from 60^0 to 90^0. This effect has never, to the author's knowledge, been included in an atmospheric IR radiation model. Since the

IR radiation is nearly isotropic (except in the 8-12 μ window region), substantial portions of it approach the ground at angles of 60° to 90°, and much more of this radiation is reflected than a constant ε_ν model would indicate. The present model will incorporate a typical angular dependence $e(\mu)$ for non-conductor emissivities and use

$$\varepsilon_\nu(\mu) = \varepsilon_\nu^{(0)}\, e(\mu) \tag{5.41}$$

where $\varepsilon_\nu^{(0)}$ is the hemispherical emissivity for the surface in question.

The directional-hemispherical reflectivity $A_\nu(\mu)$, the bidirectional reflectivity $\rho_\nu(\mu,\mu_r,\phi-\phi_r)$, and its azimuthal average $\bar{\rho}_\nu(\mu,\mu_r)$ were defined in Section 5.2.1. They are related as follows:[21]

$$A_\nu(\mu) = \frac{1}{\pi}\int_0^{2\pi} d\mu_r \int_0^1 d\mu_r\, \mu_r\, \rho_\nu(\mu,\mu_r,\phi-\phi_r)$$

$$= 2\int_0^1 d\mu_r\, \mu_r\, \bar{\rho}_\nu(\mu,\mu_r) \quad . \tag{5.42}$$

Kirchoff's Law allows us to relate A_ν and ε_ν,

$$A_\nu(\mu) + \varepsilon_\nu(\mu) = 1 \tag{5.43}$$

which holds without restrictions.[21] Hence, if the reflection is diffuse so that Eq. (5.31) applies for the reflected radiation, the surface boundary condition may be formulated entirely in terms of $\varepsilon_\nu(\mu)$,

$$\bar{I}_\nu(0,\mu) = \varepsilon_\nu(\mu)\, B_\nu(T_g) + 2 \int_{-1}^{0} |\mu'| \cdot \left[1 - \varepsilon_\nu(|\mu'|)\right] \bar{I}_\nu(0,\mu')\, d\mu' \quad ,$$

$$0 \leq \mu \leq 1 \quad . \tag{5.44}$$

The first (emission) term comes from the definition (5.40) of emissivity.

Only for water surfaces is the function $\bar{\rho}_\nu$ obtainable in the IR. It is theoretically calculable as discussed in Section 5.2.1. From it we may obtain $A_\nu(\mu)$, and hence $\varepsilon_\nu(\mu)$, according to Eqs. (5.42) and (5.43). For such surfaces, we will replace the second term in Eq. (5.44) by the exact result Eq. (5.33), eliminating the assumption of diffuse reflection.

The boundary condition on the terrestrial radiation at the top of the atmosphere is homogeneous:

$$\bar{I}_\nu(z_o,\mu) = 0 \quad , \quad -1 \leq \mu \leq 0 \quad . \tag{5.45}$$

5.3 DISCRETIZATION OF THE TRANSPORT EQUATION FOR NUMERICAL SOLUTION

The intensity $\bar{I}_\nu$ is a function of the three independent variables ν, z, and μ. The discretization of each of these variables in turn is discussed in Sections 5.3.1 through 5.3.3. The actual numerical solution of the transport equation is treated in Section 5.3.4.

5.3.1 ν-Discretization

In the regions of the spectrum in which line absorption is important (which includes most of the terrestrial

radiation spectrum except for the 8-12μ "window" region and the solar spectrum below 0.3μ and above 1μ) the absorption coefficient α_ν varies extremely rapidly with frequency. So, therefore, will the intensity, making it infeasible to solve for $\bar{I}_\nu$ at a set of discrete ν's because of the large number of ν's that would need to be taken.

Instead, we shall solve for the frequency-averaged intensities:

$$I_i(z,\mu) = \frac{1}{\nu_{i+1} - \nu_i} \int_{\nu_i}^{\nu_{i+1}} \bar{I}_\nu(z,\mu)\, d\nu$$

over an appropriately small number of frequency intervals (ν_i,ν_{i+1}). Because of this, the integro-differential form (5.9) of the transport equation is unsuitable. (It is not known how to approximate the integral

$$\int_{\nu_i}^{\nu_{i+1}} \alpha'_\nu \bar{I}_\nu\, d\nu ,$$

in which both α'_ν and $\bar{I}_\nu$ are violently oscillating, in terms of I_i.) We must use the integral form (see Eq. (5.2))

$$\bar{I}_\nu(z,\mu) = \bar{I}_\nu(0,\mu) \exp\left[-\frac{1}{\mu}\int_0^z \kappa_\nu(z')\, dz'\right] + \frac{1}{\mu}\int_0^z J_\nu(z'',\mu) \exp\left[-\frac{1}{\mu}\int_{z''}^z \kappa_\nu(z')\, dz'\right] dz'' \qquad (5.46)$$

for $\mu \geq 0$, and

$$\bar{I}_\nu(z,\mu) = \bar{I}_\nu(z_o,\mu)\ \exp\left[-\frac{1}{\mu}\int_{z_o}^{z}\kappa_\nu(z')\,dz'\right]$$

$$+\frac{1}{\mu}\int_{z_o}^{z} J_\nu(z'',\mu)\ \exp\left[-\frac{1}{\mu}\int_{z''}^{z}\kappa_\nu(z')\,dz'\right]dz'' \qquad (5.47)$$

for $\mu \leq 0$. The source function is

$$J_\nu(z,\mu) = \alpha'_\nu(z)\ B_\nu(T(z)) + \hat{S}_\nu(z,\mu)$$

where

$$\hat{S}_\nu(z,\mu) = \frac{\beta_\nu(z)}{2}\int_{-1}^{1}\bar{P}_\nu(z,\mu,\mu')\ \bar{I}_\nu(z,\mu')\,d\mu'$$

for the terrestrial spectrum, and

$$\hat{S}_\nu(z,\mu) = \frac{\beta_\nu(z)}{2}\left[\int_{-1}^{0}\bar{P}_\nu(z,\mu,\mu')\ \bar{I}_\nu^{diff}(z,\mu')\,d\mu'\right.$$

$$+\int_{o}^{1}\bar{P}_\nu(z,\mu,\mu')\ \bar{I}_\nu(z,\mu')\,d\mu'$$

$$\left.+\frac{S_\nu}{2\pi}\bar{P}_\nu(z,\mu,\mu_{sun})\ \exp\left\{\frac{1}{\mu_{sun}}\int_{z}^{z_o}\kappa_\nu(z')\,dz'\right\}\right]$$

for the solar spectrum. (For the solar spectrum, $\bar{I}_\nu$ should also be replaced by $\bar{I}_\nu^{diff}$ in Eq. (5.47) .) Applying the frequency averaging operator

$$\frac{1}{\nu_{i+1} - \nu_i} \int_{\nu_i}^{\nu_{i+1}} d\nu$$

to both sides of Eq. (5.46) ,

$$I_i(z,\mu) = I_i(0,\mu)\ T_i(\mu,0,z)\ \Sigma_i\ (\mu,0,z)$$

$$+ \int_0^z B_i(T(z''))\ \Sigma_i\ (\mu,z'',z)\ \frac{\partial T_i(\mu,z'',z)}{\partial z''}\ dz''$$

$$+ \frac{1}{\mu} \int_0^z \hat{S}_i(z'',\mu)\ \Sigma_i\ (\mu,z'',z)\ T_i\ (\mu,z'',z)\ dz'' \qquad (5.48)$$

where by definition,

$$\Sigma_i\ (\mu,z_1,z_2\ ) \equiv \exp\left[-\frac{1}{\mu} \int_{z_1}^{z_2} \beta_i(z)\ dz\right] \qquad (5.49)$$

$$T_i(\mu,z_1,z_2) \equiv \frac{1}{\nu_{i+1} - \nu_i} \int_{\nu_i}^{\nu_{i+1}} \exp\left[-\frac{1}{\mu} \int_{z_1}^{z_2} \alpha'_\nu(z)dz\right] d\nu \qquad (5.50)$$

and

$$\hat{S}_i(z,\mu) = \frac{\beta_i(z)}{2} \int_{-1}^{1} P_i(z,\mu,\mu')\ I_i(z,\mu')\ d\mu' \qquad (5.51)$$

for the terrestrial spectrum, while

$$\hat{S}_i(z,\mu) = \frac{\beta_i(z)}{2}\left[\int_{-1}^{0} P_i(z,\mu,\mu')\ I_i^{diff}\ (z,\mu')\ d\mu'\right.$$

$$+ \int_0^1 P_i(z,\mu,\mu')\ I_i(z,\mu')\ d\mu' \qquad (5.52)$$

$$\left. + \frac{\bar{S}_i}{2\pi} P_i(z,\mu,\mu_{sun})\ \Sigma_i\ (\mu_{sun},z_o,z)\ T_i(\mu_{sun},z_o,z)\right]$$

for the solar spectrum. $\bar{S}_i$ is taken as the frequency-averaged solar flux

$$\bar{S}_i = \frac{1}{\nu_{i+1} - \nu_i} \int_{\nu_i}^{\nu_{i+1}} S_\nu\ d\nu$$

since this is the form in which solar spectral data is always presented. The quantities β_i , B_i , P_i are the corresponding quantities β_ν , B_ν , $\bar{P}_\nu$ evaluated at the mid-point $\bar{\nu}_i$ of the frequency interval,

$$\bar{\nu}_i = \tfrac{1}{2}(\nu_i + \nu_{i+1}) \quad .$$

The T_i of Eq. (5.50) are called transmission functions.

An important approximation has had to be made in deriving Eq. (5.48), aside from the relatively trivial one of approximating the slowly varying (in frequency) functions B_ν, β_ν, and $\bar{P}_\nu$ by their values at the mid-point. It is the commuting of the frequency-average with $\bar{I}_\nu(0,\mu)$, and with $\bar{I}_\nu(z,\mu')$ in the scattering term, and the replacement of these quantities by $I_i(0,\mu)$ and $I_i(z,\mu')$. In view of boundary conditions such as Eq. (5.44), the boundary intensities $\bar{I}_\nu(0,\mu)$ may be rapidly varying functions of ν (unless $\epsilon_\nu \equiv 1$). Similarly, in the presence of substantial line absorption, $\bar{I}_\nu(z,\mu')$ will vary rapidly with ν. This hurdle has resulted in two divergent bodies of independent research in atmospheric radiation; one school neglects scattering (e.g., Kyle, Ref. 26), the other neglects absorption (e.g., Sekera, Ref. 27). The primary thrust of the first school is the accurate calculation of the transmission functions T_i. Once these are known, and $\hat{S}_i = 0$, $\Sigma_i = 1$, the numerical integration of Eq. (5.48) is almost trivial. The second school concerns itself with techniques for solving the integral equation (5.48) when $T_i = 1$, $B_i = 0$ (in which case the commutation of the frequency average with $\bar{I}_\nu(0,\mu)$ and $\bar{I}_\nu(z,\mu')$ is a valid approximation). To the author's knowledge, no one in all the vast literature on atmospheric radiation has considered simultaneously line absorption and scattering.

The reason for this apparent lacuna is that, over large portions of the solar and terrestrial spectrums, either scattering or line absorption is dominant. They might only

by comparable in magnitude in the near infrared region (1-5μ) where there are some weak H_2O bands, and in parts of the 8-12μ window region. Therefore, the error that we make in doing the frequency-average of the scattering term in Eq. (5.46) will tend to be large in only a small fraction of the frequency intervals; presumably these errors will have little impact on the frequency-integrated flux of Eq. (5.14).

A potentially serious approximation is the replacement of $\bar{I}_\nu(0,\mu)$ by $I_i(0,\mu)$, particularly in the strong IR absorption bands. If, in Eq. (5.44), the emissivity indeed falls to zero over a span of angles of 30^0 or so, then the second term in that equation is not negligible. Since the incident intensities $\bar{I}_\nu(0,\mu')$ in that term will be rapidly varying functions of ν, so will the reflected intensities, and hence $\bar{I}_\nu(0,\mu)$ itself. A mitigating circumstance in favor of the approximation is that, in the strong IR bands, the surface boundary condition will become unimportant after about the first kilometer or two; that is, the transmission function $T_i(\mu,0,z)$ multiplying $I_i(0,\mu)$ becomes negligibly small for $z > 1\text{-}2$ km. Nevertheless, one would hope to do the boundary layer correctly. Therefore, ways to skirt this difficulty are actively being sought.

One possible avenue of approach is to revert to the integro-differential form (5.9) and attempt to deal with the integral

$$\frac{1}{\nu_{i+1} - \nu_i} \int_{\nu_i}^{\nu_{i+1}} \alpha'_\nu \, \bar{I}_\nu \, d\nu$$

(the frequency-averages of the other terms are trivial).

Defining

$$\bar{\alpha}_i \equiv \frac{\int_{\nu_i}^{\nu_{i+1}} \alpha_\nu \bar{I}_\nu \, d\nu}{\int_{\nu_i}^{\nu_{i+1}} \bar{I}_\nu \, d\nu} \tag{5.53}$$

the integral may be approximated

$$\frac{1}{\Delta\nu} \int_{\nu_i}^{\nu_{i+1}} \alpha'_\nu \bar{I}_\nu \, d\nu = \left(1 - e^{-h\bar{\nu}_i/kT}\right) \bar{\alpha}_i \, I_i$$

$\bar{\alpha}_i$ is a function of both z and μ, in general. It also depends on the intensity, of course, which is the source of the difficulty. Nevertheless, by calculating $\bar{\alpha}_i$ for various typical intensity fields in the atmosphere (obtained by detailed calculations) regularities might emerge which would allow us to pick a universal $\bar{\alpha}_i(\mu,z)$. If this were possible, it would not only alleviate the difficulties discussed above but would actually be simpler to tabulate than T_i, which depends on three arguments.

The computation of the transmission functions T_i has a long history. The earliest attempts were based on band models, in which simple analytical representations of line strengths, positions, and shapes were assumed. As accurate line-by-line absorption data has become available,[28] both from theory and experiment, transmission function computations have incorporated it. Such detailed line-by-line transmission function computations are incredibly expensive in terms of computer time. Considering that sometimes integration steps as small as 0.001 cm^{-1} must be taken[29], and that the region

of significant absorption extends from 10,000 cm^{-1} (1μ) to 250 cm^{-1} (40μ), with only a few gaps, the magnitude of the problem becomes apparent. As an example, Kyle[29] used 15 minutes of CDC 6600 time to compute transmission functions between a single pair of atmospheric levels z_1 and z_2 for a single value of μ, and for the wavelength interval 1.7μ - 20μ. Multiply this by the number of angles N_μ and by the number of pairs of levels $\frac{1}{2}N_z(N_z-1)$, and the computing time to obtain a complete set of transmission functions becomes truly formidable (for 6 angles and 15 levels it would be 157½ hours). Fortunately, transmission functions are not terribly sensitive to the temperature profile*, and so could be tabulated once and for all for various standard profiles (tropical, mid-latitude, polar, for summer and winter, for example). This would restrict us, however, to always using the same pressure levels and angles, which could be a large disadvantage.

In view of the expense, in terms of both human and computer time, of generating transmission functions from scratch, we have decided to take advantage of the scheme of McClatchey, et.al.,[30] for generating these functions. It falls into the category of an empirical fit to known data, and is the most sophisticated in a long line of such empirical fits.[31][32] McClatchey has used detailed line-by-line absorption data to compute transmission functions for 20 cm^{-1} intervals, then has fit them with empirical functions $f^{(k)}$ such that

$$T_i^{(k)}(\mu, z_1, z_2) = f_i^{(k)}\left(\Delta w_{12}^{(k)}\right) .$$

---

* Kyle, private communication.

The superscript k refers to molecular species; there are separate f's for the uniformly mixed absorbers (CO_2, N_2O, CH_4, CO, O_2, N_2), for water vapor, and for ozone. The single argument $\Delta w_{12}^{(k)}$ is an attempt to sum up the information contained in μ, z_1, and z_2 into a single "effective absorber amount" along the slant path in question. It is calculated according to empirical prescriptions which best fit the real data; the variable mixing ratios of H_2O and O_3 are taken into consideration in these prescriptions. The total transmission function is the product of the individual ones

$$T_i(\mu, z_1, z_2) = f_i^{(1)}\left(\Delta w_{12}^{(1)}\right)\ f_i^{(2)}\left(\Delta w_{12}^{(2)}\right)\ f_i^{(3)}\left(\Delta w_{12}^{(3)}\right)$$

which is an approximation also, but an excellent one according to several authors.[33][34] The functions $f^{(k)}$ are tabulated in a subroutine called LOWTRAN which we have obtained from McClatchey and implemented on our computer.

While McClatchey's transmission functions will be used, it would be desirable in the longer range to develop a code which could generate its own transmission functions directly from the raw absorption data. Among the advantages of this are:

(1) frequency intervals $\Delta\nu$ could be chosen as desired.

(2) improvements in the raw absorption data could be readily incorporated; and

(3) the approximation of an "effective absorber amount" could be replaced by a more accurate one (the Curtis-Godson approximation[35] for example).

We have already had discussions with both Kyle and McClatchey about the possibility of obtaining their absorption data and transmission function generators. They have both mentioned the high cost in terms of computer time to generate transmission functions from scratch, but in the interests of a definitive radiation calculation we believe these costs would be justified. However, for the present time we have deferred these discussions in order to pursue other aspects of the code development.

5.3.2 μ-Discretization

The angular variable μ appears both as a parameter and as a variable of integration in the transport equation (5.9) or Eqs. (5.46) and (5.47). Because measurements and theoretical computations all indicate that the terrestrial radiation intensity and the diffuse part of the solar intensity (I_i^{diff}) are fairly smooth functions of angle, these intensities may be represented by their values at a relatively few angles μ_i, $i = 1, \ldots, N_\mu$. In order to do the flux integrals, Eq. (5.11), the intensity will be assumed to vary in a piecewise-linear fashion ($I = I^{(0)} + \mu I^{(1)}$) between the points μ_i at which it is calculated. To ensure consistency, the scattering source integrals

$$\hat{S}_i(z_j,\mu_k) \equiv \hat{S}_{ijk} = \int_{-1}^{1} P_i(z_j,\mu_k,\mu')\, I_i(z_j,\mu')\, d\mu'$$

will be done under the same assumption,

$$\hat{S}_{ijk} = \sum_{n=1}^{N_\mu - 1} \int_{\mu_n}^{\mu_{n+1}} P_i(z_j,\mu_k,\mu')(I^{(0)}_{i,j,n} + \mu' I^{(1)}_{i,j,n})\, d\mu'$$

$$= \sum_{n=1}^{N_\mu - 1} \left[q^{(0)}_{ijk,n}\, I^{(0)}_{i,j,n} + q^{(1)}_{ijk,n}\, I^{(1)}_{i,j,n} \right]$$

where the moments $q^{(\alpha)}$ of the phase function are defined as

$$q^{(\alpha)}_{ijk,n} = \int_{\mu_n}^{\mu_{n+1}} \mu^\alpha P_i(z_j,\mu_k,\mu)\, d\mu \quad .$$

The moments $q^{(0)}$, $q^{(1)}$ are to be computed and stored before the calculation of the intensities begins. It is perhaps worth noting that P_i will have to be computed over a finer angular mesh than μ_i in order to ensure accuracy in the numerical integrations leading to $q^{(0)}$, $q^{(1)}$.

An exception to the statement that I_i^{diff} varies smoothly in angle occurs when there is substantial aerosol scattering. Aerosol scattering notoriously produces a strong forward peak in the scattered intensity. This forward peak is almost as troublesome numerically as the solar beam itself (which we eliminated by the splitting of Eq. (5.34)) because it necessitates a dense mesh of angles around the solar angle μ_{sun} . Therefore, we shall use a method, tested by Hansen,[36] in which the radiation scattered into a narrow forward cone, say $\pm 2^0$ about the forward direction, is regarded as unscattered. Mathematically, this amounts to truncating the forward peak from the phase function $P_\nu(z,\mu_s)$ and decreasing the scattering coefficient β_ν accordingly. For dust and haze, which are optically thin, this seems like a reasonable procedure; in clouds, on the other hand, there might be a cumulative error after many scatterings which is not small. Hansen shows this not to be the case, however, and so the method is fully viable for all types of aerosols.

5.3.3 z-Discretization

Pressure coordinates p_j will be substituted for elevations z_j in the model by making the hydrostatic

assumption,

$$dp = -\rho g \, dz \quad .$$

This will facilitate comparison with the GCM's and agrees with general meteorological practice.

There is no discretization of the pressure coordinate which is ideal in all regions of the spectrum. The fundamental criterion for vertical zoning is that the source function J_ν may not change substantially from zone to zone. If it did, interpolations necessary to do the J_ν integrals in Eqs. (5.46) and (5.47) would be too inaccurate. In the infrared, this means that $B_\nu(T)$ (and therefore T) and the transmission function T_i may not change substantially between any two zone centers. Since 6^0/km is a typical lapse rate, zones should probably not exceed 2 km in width. Kyle, in his IR model[26], used 15 2-km levels surmounted by a sixteenth level from 30 km to the top of the atmosphere. In most of the solar spectrum, on the other hand, a much coarser vertical resolution could be tolerated, since, except within aerosol layers, the scattering source varies considerably less rapidly with height than $B_\nu(T)$.

These considerations suggest that a dynamic assignment of zoning structure would be highly desirable, based on an examination both of the source function and the spectral interval involved. In the case of the scattering source function, which involves the intensity, we would examine the scattering coefficient β_ν and phase function $\bar{P}_\nu$ instead. In absorption-dominated spectral intervals, we will zone so that the relative change in temperature T and transmission function T_i from zone to zone will be bounded. In scattering-dominated spectral intervals, we will zone so that the relative changes in β_ν and $\bar{P}_\nu$ (at selected pairs of

angles μ , μ') are similarly bounded. In spectral intervals where absorption is comparable to scattering, both T , T_i and β_ν , $\overline{P}_\nu$ will be required to be bounded in their zone-to-zone variations.

Aerosol layers, particularly clouds, will be much better resolved by such a zoning scheme than they would be by problem-independent schemes (e.g., fixed 2-km levels). By the same token, zones will not be wasted in regions across which very little is happening to the radiation field.

The problem with such a dynamic zoning scheme is that the fluxes F_i in different frequency intervals i will not be calculated at the same levels. By the very nature of the dynamic zoning scheme, however, the individual fluxes $F_i(z_k)$ will not vary greatly between z_j and z_{j+1} ; therefore, a polynomial interpolation scheme (a parabolic fit, for example) can furnish $F_i(z)$ at any intermediate level between z_j and z_{j+1} . Hence, the various fluxes $F_i(z_k)$ can be reduced to common levels $\hat{z}_k$ prior to summation over frequency,

$$F(\hat{z}_k) = \sum_i F_i(\hat{z}_k) \quad .$$

This dynamic zoning scheme is compatible with McClatchey's transmission function method discussed in Section 5.3.1. It would also be compatible with exact transmission functions, precomputed for specific levels $z_j^{(0)}$, for such functions would vary smoothly enough with level that they could be interpolated to the levels of interest.

5.3.4 Numerical Solution

The basic method for the numerical solution of the transport equation will be discrete ordinates with scattering iteration.

The discrete ordinates method was first proposed by Chandrasekhar.[37] It involves fixing the angle, $\mu = \mu_i$, in the integral form (5.46) and (5.47) of the transport equation and integrating from one boundary to the other in discrete steps Δz. The method is simply ray-tracing, accounting for sources and sinks along the ray trajectory (and, incidentally, ignoring the slight bending of the ray due to refraction).

The Planck function $B_\nu(T(z))$ in Eqs. (5.46) and (5.47) is to be interpolated linearly in z-space between points z_i at which it is known, for purposes of doing the z-integrations in (5.46) and (5.47) numerically. This ensures that the diffusion limit of radiative transport will be recovered. To see this, suppose there is no scattering, that $\kappa_\nu = \alpha'_\nu$ is independent of z, and that we are sufficiently many optical mean free paths from the boundary that the boundary term in Eq. (5.46) is negligible. Then

$$\bar{I}_\nu(z,\mu) = \frac{1}{\mu}\int_0^z \alpha'_\nu\, B_\nu\Big(T(z'')\Big)\, \exp\left[-\frac{\alpha'_\nu}{\mu}(z-z'')\right] dz'' \quad .$$

A partial integration leads to

$$\bar{I}_\nu(z,\mu) = B_\nu\Big(T(z)\Big) - B_\nu\Big(T(0)\Big)\exp\left[-\frac{1}{\mu}\alpha'_\nu z\right] - \int_0^z \frac{\partial B_\nu}{\partial z''}\exp\left[-\frac{1}{\mu}\alpha'_\nu(z-z'')\right]dz'' \quad .$$

We neglect the second term on the right-hand side because the optical path from the boundary $\alpha'_\nu z$ is assumed large. In the third term, if B_ν is linear in z'', then $\partial B_\nu/\partial z''$

is constant, and

$$\bar{I}_\nu(z,\mu) = B_\nu\big(T(z)\big) - \frac{\mu}{\alpha'_\nu}\frac{\partial B_\nu}{\partial z} \quad .$$

This is the diffusion approximation.

The scattering iteration process can best be explained by writing Eq. (5.46) schematically as

$$\bar{I}_\nu = I_o + L(\bar{I}_\nu) \quad .$$

Into I_o we have lumped all the known terms (the ones not involving $\bar{I}_\nu$). L is a double integral operator operating on $\bar{I}_\nu$ and $L(\bar{I}_\nu)$ is the scattering source term. The iteration we shall use to solve this equation is simply

$$\bar{I}_\nu^{(n+1)} = I_o + L(\bar{I}_\nu^{(n)})$$

where the initial iterate is

$$\bar{I}_\nu^{(0)} = I_o \quad .$$

This process is known to always converge.[37] Furthermore, when there is a significant amount of absorption the convergence is rapid. On the other hand, if absorption is negligible, as for visible radiation in a water cloud, and there are many mean free paths of scattering, convergence is painfully slow. This can, to some extent, be circumvented by initializing the intensity in the scattering-thick region not from I_o, but from one of the two famous analytic approximations which are attached to the names of Eddington and Schwarzschild, respectively.[38] A variant of this technique has, in fact, been successfully applied at S³.[39]

The radiative transfer code will be included in a realistic boundary layer code that has been developed under this contract. This boundary layer code is described in Appendix E. The coupled code will provide an opportunity to evaluate radiative effects in comparison to ordinary thermodynamic effects (latent heat transfer, etc.) in the atmosphere.

6. FUTURE STUDIES

This section of the report will attempt to outline further code development, modifications, and application of these codes to test problems which will be attempted during the next six months of this research contract.

6.1 HAIFA CODE DEVELOPMENT AND APPLICATIONS

Investigations now in progress will be completed. These include:

(1) Further test and modification of the version of HAIFA incorporating compressibility,

(2) Further test of the version of HAIFA incorporating moisture effects.

(3) Further modification of the Poisson solver to include non-cyclic inlet and outlet boundary conditions.

Several modifications to the HAIFA codes will be made in order to study phenomena which are not yet understood completely. Among the major code developments will be:

(1) the addition of a turbulence scheme to the basic HAIFA code, and

(2) the addition of Coriolis terms to determine their importance in mountain wave problems on the meso-scale.

It is also desirable to take account of effects in three spatial dimensions. It will not be possible, however, to develop a code and carry out such calculations under the current contract; the scope and magnitude of the problem will be investigated for future consideration.

Additional calculations to be performed during the remaining period of the current contract include:

(1) Investigation of the effect of mountain shape on lee waves, and

(2) Calculation of a problem using a fully compressible code for comparison with a standard problem calculated with HAIFA.

These results and those from previously calculated cases will be analyzed more quantitatively to characterize the momentum and energy transports. As a part of this analysis, the perturbed pressure field will be calculated through solution of the governing Poisson equation.

The final two months of the contract period will be spent in analyzing and attempting to parameterize the results of the many calculations in such a manner as to be useful in the global circulation model calculations.

REFERENCES

1. Crowley, W.P., "Numerical Advection Experiments," Monthly Weather Review 1, pp. 1-11, January 1968.

2. Buzbee, B.L., F.W. Door, J.A. George, and G.H. Golub, "The Direct Solution of the Discrete Poisson Equation on Irregular Regions," STAN-CS-71-195, December 1970.

3. Cooley, J.W. and J.W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Mathematics of Computation 19, pp. 297-301 (1965).

4. Crowley, W.P., "A Global Numerical Ocean Model; Part 1," J. Comp. Phys. 3, pp. 111-147 (1968).

5. Ogura, Y. and N.A. Phillips, "Scale Analysis of Deep and Shallow Convection in the Atmosphere," J. Atmos. Sci. 19, pp. 173-179 (1962).

6. Orville, H.D., "Ambient Wind Effects on the Initiation and Development of Cumulus Clouds Over Mountains," J. Atmos. Sci., pp. 385-402, May 1968.

7. Liu, J.Y. and H.D. Orville, "Numerical Modeling of Precipitation and Cloud Shadow Effects on Mountain-Induced Cumuli," J. Atmos. Sci. 6, pp. 1283-1298, November 1969.

8. Kessler, E., "On the Distribution and Continuity of Water Substance in Atmospheric Circulation," Meteor. Monographs, No. 32, November 1969.

9. Srivastava, R.C., "A Study of the Effect of Precipitation on Cumulus Dynamics," J. Atmos. Sci. 2, pp. 36-45 (1967).

10. Arnason, G., R.S. Greenfield, and E.A. Newburg, "A Numerical Experiment in Dry and Moist Convection Including the Rain Stage," J. Atmos. Sci 25, pp. 404-415 (1968).

11. Palm, E. and A. Foldvik, "Contribution to the Theory of Two-Dimensional Mountain Waves," Geofysiske Publikaszoner Geophysica Norvegrca 6 (1959).

12. Lilly, D.K., private Communication with M.G. Wurtele (1971).

REFERENCES, contd.

13. Foldvik, A. and M.G. Wurtele, "The Computation of the Transient Gravity Wave," Geophys. J.R. Astr. Soc. 13, pp. 167-185 (1967).

14. Lyra, G., "Theory der Stationäre Leewellen-strömung in freier Atmosphare," Z. Angew. Math. Mech. 23, pp. 1-28 (1943).

15. Penndorf, R., J. Opt. Soc. Amer. 47, pp.176 (1957).

16. Kerker, M., "The Scattering of Light and Other Electromagnetic Radiation," Academic Press, New York, pp. 584 (1969).

17. Howell, H. and H. Jacobowitz, J. Atmos. Sci. 27, pp. 1195 (1970).

18. Goody, R., "Atmospheric Radiation," Oxford Univeristy Press (1964).

19. Thekaekara, M., "The Solar Constant and the Solar Spectrum Measured from A Research Aircraft," NASA Technical Report No. TR-R-351 (1970).

20. Paltridge, T., J. Geophy. Res. 76, pp. 2857 (1971).

21. Siegel, R. and J. Howell, "Thermal Radiation Heat Transfer, Vol. I," NASA Special Publication SP-164 (1968).

22. Chow, M., "Calculation of IR Flux at the Sea Surface," NYU Geophysical Sciences Lab. Report No. TR-71-1 (1971).

23. Cox, C. and W. Munk, J. Opt. Soc. Amer. 44, pp. 838 (1954).

24. Kondrat'yev, K., "Radiation in the Atmosphere," Academic Press, New York (1969).

25. Sparrow, C. and D. Cess, "Radiation Heat Transfer," Brooks/Cole, Belmont, Calif. (1966).

26. Kyle, T. et.al., "Flow of Radiation in the Earth's Atmosphere," AFCRL-70-0415 (1970).

27. Sekera, Z., JQRST 8, pp. 17 (1968).

REFERENCES, contd.

28. Calfee, R., "Atmospheric Spectroscopy," Final Report ARPA Order No. 1390, Program Code 9E50, NOAA, Boulder, Colo. (1970).

29. Kyle, T., JSRT 9, pp. 1477 (1969).

30. McClatchey, R. et.al., "Optical Properties of the Atmosphere," AFCRL-70-0527 (1970).

31. Davis, P. and W. Viezee, J. Geophys. Res. 69, pp. 3785 (1964).

32. Smith, W., "A Polynomial Representation of Carbon Dioxide and Water Vapor Transmission," ESSA Technical Report No. NESC-47 (1969).

33. Kondrat'yev, K. and Y. Timofeev, "Numerical Experiments on Studying Transmission Functions of Atmospheric Gases," WMO Technical Note No. 104 (1968).

34. Hoover, G., C. Hathaway, and D. Williams, Appl. Optics 6, pp. 481 (1967).

35. Armstrong, B., J. Atmos. Sci. 25, pp. 312 (1968).

36. Hansen, J., J. Atmos. Sci. 26, pp. 478 (1969).

37. Chandrasekhar, S., "Radiative Transfer," Dover, New York (1960).

38. Menzel, D., ed., "Papers on Transfer of Radiation," Dover, New York (1966).

39. Schaibly, J. and W. Wiscombe, "Improvements in the VERA System of Codes," DASA Report No. 3SCR-481-1 (1970).

APPENDIX A

DERIVATION OF BOUSSINESQ EQUATIONS

The conservation equations governing macroscopic fluid motion are frequently simplified for problems of thermal convection by introducing certain approximations which are attributed to Boussinesq. These approximations can best be summarized by

(1) fluctuations in density which appear with the advent of motion result principally from thermal (as opposed to pressure) effects, and

(2) in the conservation equations of mass and momentum, density variations may be neglected except when they are coupled to the gravitational acceleration in the buoyancy force.

These approximations are examined in the derivation of equations presented below.

The general equations of mass and momentum conservation are

$$\frac{d\rho}{dt} = -\rho\nabla\cdot\bar{V} \qquad (A.1)$$

$$\rho\frac{d\bar{V}}{dt} = -\nabla p + \nabla\cdot\bar{\bar{P}} - \rho g\hat{k} \quad . \tag{A.2}$$

For purposes of this derivation the viscous stress tensor $\bar{\bar{P}}$ will be dropped from the equations. The equation of state will be assumed to be of the form

$$\rho = \rho(p,T) \quad . \tag{A.3}$$

The basic approximation to be made may be examined by the following procedure:

(1) Let f represent any one of the state variables. It will be expressed in the following form

$$f = f_m + f_o(Z) + f'(x,z,t) \tag{A.4}$$

where

f_m = space average of f

$f_o(Z)$ = variation of f in the absence of motion

$f'(x,z,t)$ = fluctuations in f resulting from fluid motions.

(2) If a scale height is introduced as

$$H(f) = \left|\frac{1}{f_m}\frac{df_o}{dz}\right|^{-1} \tag{A.5}$$

the basic approximation is that the fluid be confined to a layer whose thickness, d, is much less than that of the scale height $(d << H)$.

In particular, Eq. (A.5) implies that $d/H(\rho) << 1$. On integrating this latter condition over the layer, one concludes that

$$\frac{\Delta\rho_o}{\rho_m} \equiv \varepsilon << 1 \quad , \tag{A.6}$$

where $\Delta\rho_o$ is the maximum variation of ρ_o across the layer.

It is also required in non-linear investigations to make the additional restriction that the motion induced fluctuations do not exceed, in order of magnitude, the static variation, i.e.,

$$\left|\frac{\rho'}{\rho_m}\right| \leq O(\varepsilon) \quad . \tag{A.7}$$

Condition A.7 must be verified a posteriori from solutions of the problem. In the absence of motion and introducing Eq. (A.4), the vertical component of Eq. (A.2) is

$$\frac{\partial p_o}{\partial z} = -g\rho_m - g\rho_o \quad . \tag{A.8}$$

Introducing the hydrostatic relation into Eq. (A.2), we have

$$\rho\left(\frac{\partial \bar{v}}{\partial t} + \bar{v}\cdot\nabla\bar{v}\right) = -\nabla p' - g\rho' \hat{k} \quad . \tag{A.9}$$

We may introduce Eqs. (A.4) and (A.6) into the continuity Eq. (A.1) to obtain

$$\nabla\cdot\bar{v} = -\left(\frac{\partial}{\partial t} + \bar{v}\cdot\nabla\right)\left(\varepsilon\frac{\rho_o}{\Delta\rho_o} + \frac{\rho'}{\Delta\rho_o}\right) + O(\varepsilon^2) \quad . \tag{A.10}$$

Hence to order ε , Eqs. (A.9) and (A.10) may be written

$$\frac{\partial \bar{v}}{\partial t} + \bar{v}\cdot\nabla\bar{v} = -\frac{1}{\rho_m}\nabla p' - g\varepsilon\frac{\rho'}{\Delta\rho_o}\hat{k} \qquad (A.11)$$

$$\nabla\cdot\bar{v} = 0 \qquad (A.12)$$

In Eq. (A.11) we have retained the term $g\varepsilon(\rho'/\Delta\rho_o)\,\hat{k}$ even though it contains ε as a factor. This procedure is necessary if we are to study convection problems in the Boussinesq approximation, and the following justification may be made: The quantity $\frac{\partial \bar{v}}{\partial t}$ measures the characteristic acceleration of the fluid. Now the system is driven by fluctuations of the density field, and hence we must insist that the characteristic acceleration be of order $(g\varepsilon\rho'/\Delta\rho_o)$. This, in turn, forces the conclusion that the acceleration of gravity is always much greater than the characteristic acceleration, i.e.,

$$\frac{g\rho'}{\Delta\rho_o} \sim O\left(\frac{1}{\varepsilon}\frac{\partial \bar{v}}{\partial t}\right) .$$

APPENDIX B

WATER PRODUCTION TERM

The water production term consists of three physical phenomena which can add to, subtract from, or change the state of the water in the atmosphere. Thus the production term P_r is written as the sum of (1) the evaporation of rainwater outside of the cloud, $\beta(r-r_s)$; (2) the conversion of cloud water to rainwater; and (3) the accretion of cloud water by rainwater, S_a ;

$$P_r = \beta(r-r_s) + S_c + S_a \qquad (b.1)$$

where β is the evaporation parameter and is assumed constant,

S_c is a linear function of the cloud water content; and

S_a is a variable dependent on both the cloud water content and the rainwater content.

The expressions for S_c and S_a , taken from Orville[7] are:

$$S_c = \alpha(\ell_c - \ell_{cr}) \qquad (B.2)$$

$$S_a = 4.6 \times 10^{-3}\, \ell_c(\ell_r)^{0.95} \quad . \qquad (B.3)$$

Orville has run test problems using ranges of the constant α from 10^{-4} sec^{-1} to 2×10^{-3} sec^{-1} and values of ℓ_{cr} from 0 to 5×10^{-4} gm gm^{-1} .

Variations of these parameters and their values will also be studied as part of the research on determining the moisture effects on mountain lee waves and/or the drag force on the air flow.

APPENDIX C

EDIT QUANTITIES

The momentum flux (wave drag) associated with gravity waves is fundamentally different from other known momentum transport processes like surface frictional drag inasmuch as it may act across deep atmospheric layers. Sawyer[C1] pears to be the first to have pointed out that in stratified flow the pressure is systematically higher on the upstream side, resulting in a drag force on the obstacle, and a corresponding drag of opposite sign on the air stream.

For steady flow over an obstacle, Vergeiner[C2] shows that the equation for horizontal momentum, Eq. (2.1) may be integrated over a slab between the mountain and a fixed height H to give a drag on the mountain equivalent to

$$D(\text{drag}) = -\int_{-L}^{L} (\rho uw)_{z=H}\, dx - \int_{\substack{\text{obstacle}\\ \text{height}}}^{H} (p + \rho u^2)\, dz \Bigg|_{x=-L}^{x=L} \qquad (C.1)$$

where L and $-L$ indicate lengths upstream and downstream from the obstacle. If $(p + \rho u^2)$ is the same for upstream and downstream $(L \to \infty)$, the momentum flux $\int_{-\infty}^{\infty} \rho uw\, dx$ is constant as a function of z and equal to the drag. In the linearized case, the drag may be transformed by using the linearized equation for horizontal momentum into

$$-\bar{\rho}(z) \int_{-\infty}^{\infty} u'w' \, dx = -\rho_o \int_{-\infty}^{\infty} u_o'w_o' \, dx \qquad (C.2)$$

where the o subscripts indicates values taken at the surface of the mountain.

The problem we are concerned with in the research is the calculation of a drag for the air flow over a mountain where the numerical calculations are not capable of being run to steady state values. The quantities presently being edited from the results of the numerical calculations are

$$2L\int_{-L}^{L} u'w' \, dx \qquad \text{or} \qquad \overline{u'w'}$$

where 2L is the distance of the horizontal grid. The quantities u'w' have been obtained by two different methods. Initially, an averaging method was used, i.e., u' was found from

$$u' = (u_{i,j} + u_{i+1,j})/2 - \bar{u} \qquad (C.3)$$

where

$$\bar{u} = \frac{1}{I} \sum_{i=1}^{I} (u_{i,j} + u_{i+1,j})/2 \quad .$$

v' was found as $(v_{i,j} + v_{i,j+1})/2$. In this manner, the cell-centered velocities were found and then

$$2\frac{1}{L}\int_{-L}^{L} u'v'\,dx$$

was calculated by averaging the product of the two perturbations velocities over the horizontal grid length.

The second method of finding the drag consisted of editing Crowley's second order scheme to find the vertical flux of horizontal momentum. Crowley's scheme for the momentum flux is written in finite difference form is

$$\frac{1}{2} v_{i,j+1}\left[\left(u_{i,j+1} + u_{i,j}\right) - \frac{\Delta t}{\Delta z} v_{i,j+1}\,(u_{i,j+1} - u_{i,j})\right] \quad . \qquad (C.4)$$

By averaging this quantity over the horizontal grid the result is just equal to the product of the horizontal and vertical perturbation quantities. This follows by noting that

$$\frac{1}{I}\sum_{i}^{I} v_{i,j}\,u_{i,j} = \frac{1}{I}\sum_{i}^{I} v'_{i,j}\,(u'_{i,j} + \bar{u}_{i,j}) = \overline{v'_{i,j}\,u'_{i,j}} \qquad (C.5)$$

A comparison of these two methods shows essentially identical results.

The results as described above for the problems completed to date are given in Section 4 of this report. In order to attempt to understand these results, various lengths of the horizontal average were used in obtaining values of $\overline{u'w'}$. These results are also discussed in Section 4.

APPENDIX D

HAIFA CODE LISTING

A listing of the HAIFA code which is presently operating on the UNIVAC 1108 at Systems, Science and Software (S^3) is included in this appendix. The advection scheme is second order only in this version of the code. Working versions at S^3 allow either second order or fourth order schemes.

```
S-CUBED*TPFS.MIFINC
      1       MIFINC PROC
      2       C
      3       C     **PARAMETER STATEMENT
      4       C
      5             PARAMETER MI=69,MY=35,MIP1=MI+1,MYP1=MY+1,MI=25,M2=MI/2
      6       C
      7       C     **BLANK COMMON
      8             COMMON DUM(10),IDUM(10),TIME,DT,DTFIX,TMAX,ICYCL,CYCMAX,SAVE,
      9            1   MIM1,MYM1,MIM2,MYM2,MIM3,MYM3,DX,DY,ROX,ROY,DTDX,DTDY,DV,DTP,
     10            2   TPDT,VDY,TPDX,VDX,DX2,DY2,CO,DT,DV,TDI,WDI,DS,VT,DT1,DT2,DT3,
     11            3   DT4,DDT,VTX,VTY,AXY,BXY,DXY,EMAX,N,M,PSID,VORO,CONO,TTEND,TCNO,
     12            4   PDS,VELOO,NOBS,RESTRT,ISTART,IPLOT,TPLOT,IPRINT,TPRINT,DTSO,
     13            5   IDUMP,TDUMP,TIMM,TINN,TINO,PLOT,PRINTX,DUMP,MODIFY,COUNT,
     14            6   C(MI,MI),PSIFIX(MI),IMOCX(MI,2),
     15            7   P(MIP1,MYP1),R(MX,MY),T(MX,MY),BI(MX,MY),TI(MX,MY),U(MIP1,MY),
     16            8   V(MX,MYP1),TOI(MY),RTOI(MY),TEMP(MY),SI(MX,MYP1),S2(MX,MYP1),
     17            9   RR(M2),YT(M2),SL(MX),UU(MX),VV(MX)
     18       C
     19       C
     20       C     **TYPE STATEMENTS
     21             INTEGER PSID,VORO,TTENO,TCNO,VELOO
     22             LOGICAL PLOT,PRINTX,DUMP,SAVE
     23       C
     24        END
```

```
S-CUBE0:TPFS.MAIN
      1       C     MAIFA IS A TWO DIMENSION INCOMPRESSIBLE FLUID DYNAMICS CODE UTILIZING THE
      2       C     BOUSSINESQ APPROXIMATION.
      3       C
      4       C*****************************************************************************
      5       C     ALL UNITS IN THIS CODE ARE MKS. TEMPERATURE IN DEGREES KELVIN
      6       C*****************************************************************************
      7       C
      8       C     ***************************************************************
      9       C          D E F I N I T I O N  O F  V A R I A B L E S
     10       C       NX.....NUMBER OF CELLS IN X DIRECTION
     11       C       NY.....NUMBER OF CELLS IN Y DIRECTION
     12       C       DX.....SIZE OF CELL IN X DIRECTION (METERS)
     13       C       DY.....SIZE OF CELL IN Y DIRECTION (METERS)
     14       C       TIME...CURRENT PROBLEM TIME
     15       C       TMAX...MAXIMUM TIME LIMIT FOR CALCULATION
     16       C       DT.....TIME STEP FOR CURRENT CYCLE
     17       C       ICYCL..CURRENT CYCLE NUMBER
     18       C       CYCMAX.MAXIMUM NUMBER OF CYCLES FOR CALCULATION
     19       C       P......PSI (STREAMFUNCTION)
     20       C       T......TEMPERATURE DIFFERENCE FROM INITIAL CONDITIONS
     21       C       TI.....CHANGE IN T ARRAY FOR ONE CYCLE
     22       C       W......VORTICITY
     23       C       WI.....CHANGE IN W ARRAY FOR ONE CYCLE
     24       C       U......VELOCITY IN X DIRECTION
     25       C       V......VELOCITY IN Y DIRECTION
     26       C       DV.....EDDY DIFFUSION CONSTANT FOR MOMENTUM
     27       C       DTP....EDDY DIFFUSION CONSTANT FOR TEMPERATURE
     28       C       TOI....DERIVATIVE OF ORIGINAL TEMPERATURE PROFILE W.R.T. Y
     29       C              PLUS GAMMA
     30       C       BTOI...GRAVITATIONAL CONSTANT/(2.0*DX*ORIGINAL TEMPERATURE)
     31       C       TEMP...ORIGINAL TEMPERATURE
     32       C                    OUTPUT FLAGS
     33       C       IPRINT.PRINTER EDIT ON CYCLES
     34       C       TPRINT.PRINTER EDIT ON TIME
     35       C       IPLOT..CONTOUR PLOTS ON CYCLES
     36       C       TPLOT..CONTOUR PLOTS ON TIME
     37       C       IDUMP..TAPE DUMP ON CYCLES
     38       C       TDUMP..TAPE DUMP ON TIME
     39       C                    CONTOUR PLOT FLAGS
     40       C       TTEMPO.TOTAL TEMPERATURE
     41       C       VORO...VORTICITY
     42       C       PSIO...STREAMFUNCTION
     43       C       TEMPO..TEMPERATURE DIFFERENCE FROM ORIGINAL PROFILE
     44       C       VELOO..U AND V VELOCITY
     45       C                    OPTIONS
     46       C       DTFIX..FORCE CONSTANT TIME STEP
     47       C       RESTRT.RESTART A PROBLEM FROM TAPE
     48       C       ISTART.RESTART CYCLE
     49       C       NOBS...NUMBER OF IRREGULAR POINTS IN STREAMFUNCTION
     50       C     ***************************************************************
     51
     52
     53             INCLUDE MIPINC.LIST
```

```
      C
      C
      C     **READ PROBLEM DATA
      C
            CALL INPUT
      C
      C     **SET UP SPECIFICATIONS FOR THE POISSON EQTN SOLVER
      C
            DYSQ=DY*DY
            SS=DYSQ/(DX*DX)
            CALL SETUP(NX,NYP1,S1,RR,SS,TT,SL,UU,VV)
      C
            IF(RESTRT .GT. 0.0) GO TO 20
      C
      C     **IF THERE ARE IRREGULARITIES IN THE POISSON GRID, CALL OBSET
      C
            IF(NOBS .EQ. 0) GO TO 10
            CALL OBSET
            CALL LAPLAC
      C      ** COMPUTE VELOCITIES AND PLOT
         10 CALL DEFINE
            CALL VELOC
            CALL OUTPUT
      C
      C     **DEFINE VARIABLES
      C
      C     **INITIALIZE SOME QUANTITIES
            ICYCL=0
         20 SAVE=.FALSE.
            IF(TDUMP .GT. 0.0  .OR.  IDUMP .GT. 0) SAVE=.TRUE.
      C*****************************************************************
      C                  MAIN LOOP
      C*****************************************************************
         25 ICYCL=ICYCL+1
      C     **IF THERE IS NOT ENOUGH TIME TO COMPLETE ANOTHER CYCLE, AND
      C          A DUMP OF FINAL RESULTS HAS BEEN REQUESTED, DUMP NOW
            IF(SAVE) CALL S3WARN(10,1,$7001
            DTDX=DT*RDX
            DTDY=DT*RDY
      C
      C     **COMPUTE THE NEW VORTICITY AND TEMPERATURE FIELDS DUE TO
      C         ADVECTION, DIFFUSION, AND SOURCES
      C          FIELDS DUE TO ADVECTION, DIFFUSION, AND SOURCES
      C
            CALL UPDATE
      C
      C     **SOLVE POISSON EQUATION TO OBTAIN NEW STREAMFUNCTION PSI
      C
            CALL LAPLAC
      C
      C     **COMPUTE U AND V VELOCITIES
      C
            CALL VELOC
      C
```

```
      C     **ADVANCE THE TIME
      C
            TIME=TIME+DT
      C
      C     **COMPUTE THE TIME STEP FOR THE NEXT CYCLE BASED ON WIND VELOCITY,
      C          DIFFUSION VELOCITY, AND BOUYANT RISE VELOCITY
      C
            CALL TIMSTP
      C
      C     **DETERMINE IF THIS CYCLE'S RESULTS ARE TO BE PLOTTED, PRINTED,
      C          OR DUMPED
      C
      C
            PLOT=.FALSE.
            PRINTX=.FALSE.
            DUMP=.FALSE.
            CALL PRTTST
      C
      C     **SELECT OUTPUT ROUTINES
            IF(PRINTX) CALL EDIT
            IF(PLOT) CALL OUTPUT
            IF(DUMP) CALL WTAPE
      C
      C     **IS THIS RUN COMPLETED?
      C
            IF(TIME .GE. TMAX  .AND.  TMAX .GT. 0.0) GO TO 200
            IF(CYCL .GE. CYCMAX  .AND.  CYCMAX .GT. 0) GO TO 200
      C
      C     **RETURN TO MAIN LOOP FOR ANOTHER CYCLE
      C
            GO TO 25
      C
      C     **NORMAL EXIT
      200   IF(SAVE  .AND.  .NOT. DUMP) ) CALL WTAPE
            IF(.NOT. PLOT) CALL OUTPUT
            IF(.NOT. PRINTX) CALL EDIT
            STOP NORMAL
            END
```

```
S-CUBED•TPFS•CONTUR
      1              SUBROUTINE CONTUR(IX,IY,X,K,TIXX,KK,TAB,KKK)
      2              INCLUDE HIFINC
      3              DIMENSION TAB(2),X(IX,IY)
      4              DATA NC,CMIN,CMAX,LOGFL/10,0.,0.,0/
      5        C     ••ADJUST PRINTER SPACING
      6              CALL PRTCN('M,66,0,0.')
      7              CALL PRTCN('L,2.')
      8              WRITE(6,1000) (TAB(I),I=1,2),ICYCL,TIME
      9              CALL PL(NC,X,IX,IY,IX,CMIN,CMAX,LOGFL)
     10        C     ••RESET NORMAL PRINTER SPACING
     11              CALL PRTCN('M,66,6,6.')
     12        C
     13              RETURN
     14        C
     15        C     ••FORMATS
     16        C
     17         1000 FORMAT(5X,4H••••,3X,2A6,4H••••,10X,8HCYCLE = ,I5,6X,7HTIME = ,
     18             1   F10.3)
     19              END
```

```
S-CUBED*TPFS.DEFINE
      1            SUBROUTINE DEFINE
      2            INCLUDE MIFINC
      3            RDX=1.0/DX
      4            RDY=1.0/DY
      5            DUM(1)=RDY*RDY
      6            DUM(2)=RDX*RDX
      7            DUM(3)=DX*RDY
      8            DUM(4)=DY*RDX
      9            TIMM=0.0
     10            TIMN=0.0
     11            TIMD=0.0
     12      C     **NOMINAL TIME STEP CRITERION
     13            IF(DT .LT. 1.0E-20) DT=0.01
     14      C     **GRID LIMITS
     15            NXM1=NX-1
     16            NYM1=NY-1
     17            NXM2=NX-2
     18            NYM2=NY-2
     19            NYM3=NY-3
     20            NXM3=NX-3
     21      C     **TIME STEP COEFFICIENTS
     22            DS=AMIN1(DX,DY)
     23            VT=AMAX1(DTP,DV)/DS
     24            IF(DTP+DV .LT. 1.0E-8) VT=-1.0
     25      C          DTI IS THE MAXIMUM TIME STEP FOR DIFFUSION
     26            DTI=0.8*DS/VT
     27            IF(DTI .LT. 0.0) DTI=1.0E10
     28            DT3=1.0E10
     29            DDT=1.2*DT
     30      C     **COEFFICIENTS FOR VORTICITY
     31            DY2=0.5*DY
     32            DX2=0.5*DX
     33            TPDY=DTP*DUM(1)
     34            VDY=DV*DUM(1)
     35            TPDX=DTP*DUM(2)
     36            VDX=DV*DUM(2)
     37            NDDIFF=0
     38            DUM(10)=TPDY+VDY+TPDX+VDX
     39            IF(DUM(10) .LT. 1.0E-20) NDDIFF=10
     40      C     ** CONSTANTS FOR COMPUTING DRAG COEFF.
     41            IMT=1
     42            IMIN= 100
     43            IMAX= 1
     44            AXY=0.
     45            BXY=0.
     46            DO 10  I=1,ND8S
     47            IMT=MAX0(IMT,INDEX(I,2))
     48            IMIN=MIN0(IMIN,INDEX(I,1))
     49         10 IMAX=MAX0(IMAX,INDEX(I,1))
     50            LEN =(IMAX-IMIN) * DX
     51            DXY = IMAX * 4.5
     52            DO 20  J=1,NY
     53         20 AXY= AXY +((1.225 * 298.)/(TEMP(J))) * ( P(I,J+1)-P(I,J))**2 /DY
```

```
54            BXT=((1.225*299.1/(TEMP(IMT)))*((P(I,IMT+1) - P(I,IMT))/DY)**2 *
55           2 LEN
56            PRINT 913, AXT,BXT
57        913 FORMAT(1H1,'FACTORS USED TO CALCULATE THE DRAG COEFF. (A-B)/C
58           2 A =',1PE15.5, 5X,'B =',1PE15.5)
59            RETURN
60            END
```

```
S-CUBED*TPFS.EDIT
      1              SUBROUTINE EDIT
      2              INCLUDE MIFINC
      3              WRITE(6,1000) ICYCL,TIME
      4              LINE=0
      5              DO 100 J=1,NYP1
      6              DO 90  I=1,NXP1
      7              IF(MOD(LINE,50) .LT. 0.1) WRITE(6,1100)
      8              WRITE(6,1200) J,I,P(I,J),U(I,J),V(I,J),W(I,J),T(I,J)
      9              LINE=LINE+1
     10           90 CONTINUE
     11          100 CONTINUE
     12              RETURN
     13         1000 FORMAT(    /,1X,12(10H**********),/,20X,18HSUMMARY OF RESULTS,20X,
     14             1   8HCYCLE = ,I4,20X,7HTIME = ,1PE12.5,/,1X,12(10H**********) )
     15         1100 FORMAT(1H1,3X,1HJ,4X,1HI,2X,7X,3HPSI,13X,1HU,19X,1HV,11X,
     16             1   9HVORTICITY,8X,6HA TEMP )
     17         1200 FORMAT(2X,I3,3X,I3,2X,5(2X,1PE13.6))
     18              END
```

```
S-CUBED*TPFS.EMEDIT
      1              SUBROUTINE EMEDIT
      2              INCLUDE MIPINC
      3        C     ******************************************************************
      4        C                        MOMENTUM AND ENERGY EDIT
      5        C     KINETIC ENERGY PER UNIT AREA  =(RHO/8.(*(U(I,J)+U(I+1,J))**2 +
      6        C                                     (RHO/8.)*(V(I,J)+V(I,J+1))**2
      7        C     INTERNAL ENERGY PER UNIT AREA = RHO * CPAIR * T(I,J)
      8        C     POTENTIAL ENERGY PER UNIT AREA = RHO * 9.816) * (J-.5)*DY
      9        C     MOMENTUM FLUX PER UNIT AREA = (RHO/9.)*(U(I,J)+U(I+1,J))*(V(I,J)+V(I,J+1))
     10        C     RHO = RHO(STP) *  298./(TEMP(J)+T(I,J)
     11        C     ******************************************************************
     12              DISU = 0.
     13              IPEN = DIY
     14              DISU1=0.
     15              DISU2=0.
     16              DISU3=0.
     17              IPER1=IPER*5
     18              IPER2=IPER*10
     19              IPER3=IPER*15
     20              IPE=0.
     21              IKE=0.
     22              IIE=0.
     23              IMF=0.
     24              PRINT 100, ICYCL, TIME
     25              PRINT 200
     26              PRINT 201
     27              DO 20 J=1,MY
     28              IMF=0.
     29              IKEF=0.
     30              IKE=0.
     31              IIE=0.
     32              IPE = 0.
     33              DO 10 I=1,MX
     34              RHO = (1.225 * 298.) / (TEMP(J)+ T(I,J))
     35              UI = (U(I+1,J) + U(I,J)) *.5
     36              VI =(V(I,J+1)+ V(I,J)) *.5
     37              UUVV = UI**2 + VI**2
     38              IMF = RHO * UI * VI + IMF
     39              IKEF = RHO * UUVV * VI * .5 + IKEF
     40              IKE = RHO * UUVV * .5 + IKE
     41              IIE = RHO * .24 * (TEMP(J)+T(I,J) ) + IIE
     42              IPE = IPE + RHO * 9.8 * (J-.5) + DY
     43              IF(I.EQ.IPER) DISU = DISU + RHO * UI**2
     44              IF(IPER1.EQ.I) DISU1=DISU1+RHO*UI**2
     45              IF(IPER2.EQ.I) DISU2=DISU2+RHO*UI**2
     46              IF(IPER3).EQ.I) DISU3=DISU3+RHO*UI**2
     47           10 CONTINUE
     48              PRINT 300  J, IMF, IKEF, IKE, IIE, IPE
     49              TKE = TKE +IKE
     50              TIE = TIE + IIE
     51              TMF = TMF + IMF
     52              TPE=TPE + IPE
     53           20 CONTINUE
```

```
      DISU  = DISU + DT
      DRAG = 2.0 * (ARY - DISU1/BRY
      DRAG1=2.0*(ARY-DISU1)/BRY
      DRAG2=2.0*(ARY-DISU2)/BRY
      DRAG3=2.0*(ARY-DISU3)/BRY
      TE = TKE + TIE + TPF
      PRINT 90C, TKE, TIE
      PRINT 90C, TPE, TMF
      PRINT 60C, TE,DRAG,DRAG1,DRAG2,DRAG3
  100 FORMAT(1HI,10X,'MOMENTUM AND ENERGY EDIT FOR CYCLE',I4,'  AT TIME
     2',F10.3,/)
  200 FORMAT(1X,'MOMENTUM AND ENERGIES PER ROW ARE GIVEN PER UNIT AREA
     2',///)
  201 FORMAT ( 9X,'J    MOMENTUM FLUX          KINETIC ENERGY FLUX
     2   KINETIC ENERGY          INTERNAL ENERGY       POTENTIAL ENERGY')
  300 FORMAT (1X,I4,9X,1PE12.5,13X,1PE12.5,17X,1PE12.5,12X,1PE12.5,13X,
     21PE12.5)
  900 FORMAT (1X,//,15X,'TOTAL KINETIC ENERGY =',1PE12.5,15X,'TOTAL INTE
     2RNAL ENERGY =',1PE12.5)
  90C FORMAT(15X,'TOTAL POTENTIAL ENERGY=',1PE12.5,7X,'TOTAL MOMENTUM FL
     2UX =',1PE12.5)
  60C FORMAT(29X,'TOTAL ENERGY =',1PE12.5,17X,'DRAG COEFF. =',1PE12.5)
      RETURN
      END
```

```
S>CURED.TPFS.INPUT
      1              SUBROUTINE INPUT
      2              INCLUDE MIFINC
      3        C     ********************************************************************
      4        C     G = GRAVITATIONAL CONSTANT
      5        C     GAMMA = ADIABATIC LAPSE RATE FOR AIR
      6        C     RHO   = DENSITY OF AIR  (25C -1ATM)
      7        C     TEMP RELOW INVERSION LAYER
      8        C     KT,AT1,AT2,AT3,AT4,ALPHA    T(Y)=AT1+AT2*Y**KT+AT3*Y**(KT+1)+AT4*
      9        C                                       EXP(-ALPHA*Y)
     10        C     TEMP ABOVE INVERSION LAYER
     11        C     MT,BT1,BT2,BT3,BT4,BETA     T(Y)=BT1+BT2*Y**MT+BT3*Y**(MT+1)+BT4*
     12        C                                       EXP(-BETA*Y)
     13        C       INITIAL VELOCITY PROFILE
     14        C     MT,DT1,DT2,DT3,DT4,ZETA U(TP)=MT*DT2*TP**(MT-1)+(MT+1)*DT3*TP
     15        C                                   **MT-(DT4*ZETA)*EXP(-ZETA*TP)
     16        C     ********************************************************************
     17
     18              DATA G,GAMMA,RHO/9.8,0.0),1.225/
     19        C     **INPUT NAMELIST 'START'
     20              NAMELIST /START/ RESTRT,ISTART,TMAX,CYCMAX,CHANGE
     21        C     **INPUT NAMELIST 'SPECS'
     22              NAMELIST /SPECS/ IDUMP,IPRINT,IPLOT,TPLOT,TPRINT,TDUMP,DTFIX,
     23             1    PSIO,VORO,TTEMO,VELOD
     24        C     **INPUT NAMELIST 'GEN'
     25              NAMELIST /GEN/ DX,DY,TIME,DT,BT,DV,DTP,DV,KT,AT1,AT2,AT3,AT4,NOBS,
     26             1    ALPHA,MT,BT1,BT2,BT3,BT4,BETA,YINV,MT,DT1,DT2,DT3,DT4,ZETA
     27        C
     28              READ(5,START)
     29              WRITE(6,START)
     30        C
     31        C     **IS THIS TO BE A TAPE RESTART?
     32        C
     33              IF(RESTRT .LT. 1.0E-20) GO TO 5
     34        C
     35        C*******************************************************************
     36        C                 TAPE RESTART
     37        C*******************************************************************
     38              CALL RTAPE
     39        C     **ARE CHANGES TO CONTROL VARIABLES REQUESTED?
     40              IF(CHANGE .LT. 1.0E-20) GO TO 100
     41              READ(5,SPECS)
     42              WRITE(6,SPECS)
     43              GO TO 100
     44        C
     45        C*******************************************************************
     46        C                    GENERATE PROBLEM FROM CARDS
     47        C*******************************************************************
     48            5 READ(5,SPECS)
     49              WRITE(6,SPECS)
     50              READ(5,GEN)
     51              WRITE(6,GEN)
     52        C
     53        C       SETUP INITIAL PSI, VORTICITY, AND TEMP FIELDS---FIT) ONLY
```

```
C
      DO 10  J=1,NT
      T=(J-0.5)*DT
      R(1,J)=DT2*NT*(NT-1)*T**(NT-2)+DT3*NT*(NT+1)*T**(NT-1)  + DT4*ZETA
     1   *ZETA*EXP(-ZETA*T)
      IF(TINV .LT. 1.0E-20) TINV=NTP1*DT
      IF (T.LE.TINV)TEMP(J)=AT1+(AT2*AT3*T)*T**KT+AT4*EXP(-ALPHA*T)
      IF(T.GT.TINV)TEMP(J)=BT1+(BT2+BT3*T)**NT+BT4*EXP(-BETA*T)
      TP=(J-1)*DT
      P(1,J)=DT1+(DT2+DT3*TP)*TP**NT+DT4*EXP(-ZETA*TP)
      IF(NT.EQ.0) P(1,J)=DT1+DT2+DT3*TP+DT4*EXP(-ZETA*TP)
      IF (T.LE.TINV)  GO TO 20
      GO TO 30
   20 TO1(J)=  AT2*KT*T**(KT-1) + (KT+1)*AT3*T**KT - ALPHA*AT4*EXP(-ALPH
     1   A*T) + GAMMA
      IF (KT.EQ.0) TO1(J) = AT3 + GAMMA
      GO TO 40
   30 TO1(J)=  BT2*NT*T**(NT-1) + (NT+1)*BT3*T**NT - BETA *BT4*EXP(-BETA
     1   *T)+ GAMMA
      IF (NT.EQ.0) TO1(J) = BT3 + GAMMA
   40 IF(ABS(TO1(J)).LT.1.E-6) TO1(J)=0.0
      RTO1(J)= G/12.*DX*TEMP(J))
      DO 9 I=2,NXP1
      P(I,J)=P(1,J)
    9 CONTINUE
      DO 10 I=2,NX
      R(I,J)=R(1,J)
   10 CONTINUE
C    **CASE J=NTP1
      J=NTP1
      TP=(J-1)*DT
      P(1,J)=DT1+(DT2+DT3*TP)*TP**NT+DT4*EXP(-ZETA*TP)
      DO 12 I=2,NXP1
      P(I,J)=P(1,J)
   12 CONTINUE
C
  100 RETURN
      END
```

```
$-CUBEG-TPPS.LAPLAC
      1              SUBROUTINE LAPLAC
      2        C     **THIS SUBROUTINE CONTROLS THE SOLUTION OF POISSON'S EQTN
      3        C               LAPLACIAN(PSI)=CONSTANT
      4        C
      5              INCLUDE MIPINC
      6              DIMENSION Q(NX,NYP1),QB(NX,NYP1),DELPSI(NI),QELB(NI)
      7              EQUIVALENCE (Q,S1(,(QB,S2),(DELPSI(1(,UU(1(,(DELB(1),VV(1(1
      8        C*************************************************************
      9        C*************************************************************
     10        C
     11        C     **FORM CONSTANT TERMS FOR POISSON EQTNS
     12              DO 125 I=1,NA
     13              DO 100 J=2,NY
     14              QBAR=0.25*(Q(I-1,J-1)+Q(I-1,J(+Q(I,J)+Q(I,J-1))
     15              IF(I .EQ. 1) QBAR=0.5*(Q(I,J(+Q(I-1,J(
     16              Q(I,J)=QBAR+QVSQ
     17          100 CONTINUE
     18              Q(I,1)=P(I,1)
     19              Q(I,NYP1)=P(I,NYP1(
     20          125 CONTINUE
     21              IF(NOBS .EQ. 0( GO TO 200
     22        C     **SAVE THE INITIAL Q'S IF THIS IS AN OBSTACLE PROBLEM
     23              DO 175 I=1,NA
     24              DO 175 J=1,NYP1
     25              QB(I,J)=Q(I,J)
     26          175 CONTINUE
     27        C
     28        C     **SOLVE THE UNPERTURBED GRID PROBLEM
     29        C          (THIS WILL RESULT IN ARRAY Q BEING TRANSFORMED
     30        C           TO STREAMFUNCTION VALUES)
     31        C
     32          200 CALL AYPOIS(Q)
     33        C     **ARE IRREGULAR STREAMFUNCTION POINTS IN THE GRID?
     34              IF(NOBS .EQ. 0) GO TO 800
     35        C     **COMPUTE ERROR IN OBSTACLE STREAMFUNCTION VALUES
     36              DO 300 I=1,NOBS
     37              I1=INDEX(I,1)
     38              I2=INDEX(I,2)
     39              QELPSI(I)=PSIFIX(I1-Q(I),I2)
     40          300 CONTINUE
     41        C     **COMPUTE CORRECTION TO VORTICITIES
     42              CALL MAMLTIC,QELPSI,QELQ,NOBS,NOBS,1,NI,NI(
     43        C     **MODIFY THE QB VALUES
     44              DO 400 I=1,NOBS
     45              I1=INDEX(I,1(
     46              I2=INDEX(I,2)
     47              QB(I1,I2)=QB(I1,I2)+QELQ(I)
     48          400 CONTINUE
     49        C     **APPLY POISSON EQTN SOLVER
     50              CALL AYPOIS(QB)
     51        C
     52        C     **PUT NEW STREAMFUNCTION VALUES IN THE P(*,*) ARRAY
     53        C
```

```
      DO 700 I=1,NX
      DO 700 J=2,NY
      P(I,J)=BB(I,J)
  700 CONTINUE
      GO TO 1500
C
C     **CASE WITH NO IRREGULAR GRID POINTS
C
  800 DO 900 I=1,NX
      DO 900 J=2,NY
      P(I,J)=B(I,J)
  900 CONTINUE
C
C     **SET PSI AT I=NX+1 FOR CYCLIC BOUNDARY CONDITION
C
 1500 DO 1750 J=2,NY
      PINXP1,J)=P(1,J)
 1750 CONTINUE
C
      RETURN
      END
```

```
S-CUBED*TPFS.OBSET
            SUBROUTINE OBSET
            INCLUDE HIFINC
            DIMENSION Q(NX,NYP1),RUM(NI),VX(2)
            EQUIVALENCE (Q,S(1),(RUM(1),UU(1)),(VX(1),VV(1)))
C     **MAKE INDICES GLOBAL
            INTEGER K,NOBS
            I=0
            WRITE(6,2000)
C     **READ OBSTACLE NODE INDICES AND SPECIFIED PSI VALUES
         10 I=I+1
            READ(5,1000) INDEX(I,1),INDEX(I,2),PSIFIX(I)
            WRITE(6,2010) I,INDEX(I,1),INDEX(I,2),PSIFIX(I)
            IF(I .EQ. NOBS) GO TO 50
            GO TO 10
C
C     **DEVELOP INFLUENCE MATRIX
C
         50 DO 100 K=1,NOBS
            CALL ARTZRO(Q,NX*NYP1)
            I1=INDEX(K,1)
            I2=INDEX(K,2)
            Q(I1,I2)=1.0
            CALL XYPOIS(Q)
            CALL CLOAD
        100 CONTINUE
C
C     **FORM THE INVERSE OF INFLUENCE MATRIX (STORE IN SAME MATRIX)
C
            VX(1)=1.0
            CALL GJR(C,NI,NI,NOBS,NOBS,$250,RUM,VX)
C
            RETURN
C
C     **FORMATS
C
       1000 FORMAT(2I5,1PE10.3)
       2000 FORMAT(1H1,//,10X,18HOBSTACLE NODE DATA,/,10X,1HN,9X,1HI,9X,1HJ,
          1   9X,10HPSI VALUE )
       2010 FORMAT(5X,I6,5X,I5,5X,I5,5X,1PE14.6)
C     **ERROR TERMINATION FROM GJR
        250 STOP RADINV
C*********************************************************************
C                       SUBROUTINE CLOAD
C*********************************************************************
            SUBROUTINE CLOAD
            DO 100 J=1,NOBS
            J1=INDEX(J,1)
            J2=INDEX(J,2)
            C(K,J)=Q(J1,J2)
        100 CONTINUE
            RETURN
            END
```

```
$-CURED*TPFS.ORDER2
      1              SUBROUTINE ORDER2(XX)
      2              INCLUDE HIFINC
      3              DIMENSION XX(NX,NY)
      4        C
      5        C     **STEP ONE -- X DIRECTION
      6        C
      7              CO=0.5*DTDX
      8              DO 100 J=1,NY
      9        C     **I=1 AND I=NX   USE CYCLIC BOUNDARY CONDITIONS
     10              COV=CO*U(1,J)
     11              FOLD=COV*(XX(1,J)+XX(NX,J)-2.0*COV*(XX(I,J)-XX(NX,J)))
     12              FSAVE=FOLD
     13              DO 75 I=1,NXM1
     14              COV=CO*U(I+1,J)
     15              FNEW=COV*(XX(I+1,J)+XX(I,J)-2.0*COV*(XX(I+1,J)-XX(I,J)))
     16              XX(I,J)=XX(I,J)+FOLD-FNEW
     17              FOLD=FNEW
     18           75 CONTINUE
     19        C     **CASE I=NX
     20              XX(NX,J)=XX(NX,J)+FNEW-FSAVE
     21          100 CONTINUE
     22        C
     23        C     **STEP TWO -- Y DIRECTION
     24        C
     25              CO=0.5*DTDY
     26              DO 200 I=1,NX
     27              FOLD=0.0
     28        C
     29              DO 175 J=1,NYM1
     30              COV=CO*V(I,J+1)
     31              FNEW=COV*(XX(I,J+1)+XX(I,J)-2.0*COV*(XX(I,J+1)-XX(I,J)))
     32              XX(I,J)=XX(I,J)+FOLD-FNEW
     33              FOLD=FNEW
     34          175 CONTINUE
     35        C     **CASE J=NY
     36              XX(I,NY)=XX(I,NY)+FOLD
     37          200 CONTINUE
     38              RETURN
     39              END
```

```
S-CURED*TPFS.OUTPUT
      1            SUBROUTINE OUTPUT
      2            INCLUDE HIFINC
      3            DIMENSION V1(2),V2(2),V3(2),V4(2),V5(2),V6(2)
      4            DATA (V1(I),I=1,2)  /12HU VELOCITY    /
      5            DATA (V2(I),I=1,2)  /12HV VELOCITY    /
      6            DATA (V3(I),I=1,2)  /12HA TEMP        /
      7            DATA (V4(I),I=1,2)  /12HPSI           /
      8            DATA (V5(I),I=1,2)  /12HVORTICITY     /
      9            DATA (V6(I),I=1,2)  /12HTEMPERATURE    /
     10            DATA JCYCL /-1/
     11      C     **AVOID TWO EDITS OF THE SAME CYCLE
     12            IF(ICYCL .EQ. JCYCL) RETURN
     13            JCYCL=ICYCL
     14      C     **ENERGY AND MOMENTUM EDIT
     15            CALL EMEDIT
     16            IF(ITEMO .EQ. 0) CALL CONTUR(NX,NY,T,0,TIME,0,V3,11
     17            IF(ITEMO) 9,9,7
     18          7 DO 8 I=1,NX
     19            DO 8 J=1,NY
     20          8 T(I,J) = T(I,J)+ TEMP(J)
     21            CALL CONTUR(NX,NY,T,0,TIME,0,V6,11
     22          9 IF(VORO .EQ. 0) CALL CONTUR(NX,NY,W,0,TIME,0,V5,11
     23            IF(VELO) 15,10,15
     24         10 CALL CONTUR(NXP1,NY,U,0,TIME,0,V1,11
     25            CALL CONTUR(NX,NYP1,V,0,TIME,0,V2,11
     26         15 IF(PSIO .EQ. 0) CALL CONTUR(NXP1,NYP1,P,0,TIME,0,V4,11
     27      C
     28            RETURN
     29            END
```

```
S-CUBEO.TPFS,PL
      1              SUBROUTINE PL(NNC, Z, NX, NY, NXMX, AMIN, AMAX, LOGFL)
      2              DIMENSION LQ(72), LII3II, Z(NXMX,NY)
      3              DATA LQ / '0', ' ', '1', ' ', '2', ' ', '3', ' ', '4', ' ', '5',
      4             I ' ', '6', ' ', '7', ' ', '8', ' ', '9', ' ', 'A', ' ', 'B', ' ',
      5             2 'C', ' ', 'D', ' ', 'E', ' ', 'F', ' ', 'G', ' ', 'H', ' ', 'I',
      6             3 ' ', 'J', ' ', 'K', ' ', 'L', ' ', 'M', ' ', 'N', ' ', 'O', ' ',
      7             4 'P', ' ', 'Q', ' ', 'R', ' ', 'S', ' ', 'T', ' ', 'U', ' ', 'V',
      8             5 ' ', 'W', ' ', 'X', ' ', 'Y', ' ', 'Z', ' ' /
      9              DATA ISTAR / '*' /, INAX / 131 /
     10              JMAX = .6*NY/NX*INAX + .5
     11              NC = MIN(NNC, 36)
     12              ZMIN = AMIN
     13              ZMAX = AMAX
     14              ANX = NX
     15              ANY = NY
     16              IF(ABS(ZMAX-ZMIN) .GT. 1.0E-5) GO TO 20
     17              ZMAX = Z(1,1)
     18              ZMIN = ZMAX
     19              DO 10 I = 1, NX
     20              DO 10 J = 1, NY
     21              ZMAX = MAX(ZMAX, Z(I,JI)
     22           10 ZMIN = MIN(ZMIN, Z(I,JI)
     23           20 IF (LOGFL .EQ. 0) GO TO 30
     24              IF (ZMIN .GT. 0.) GO TO 25
     25              PRINT 6020
     26         6020 FORMAT( ///// ' ***********NON-POSITIVE VALUE IN ARRAY TO
     27             I BE CONTOURED LOGARITHMICALLY.  PL IS RETURNING WITHOUT PLOTTING'I
     28              RETURN
     29           25 HZMAX = ZMAX
     30              HZMIN = ZMIN
     31              ZMAX = ALOG10(ZMAX)
     32              ZMIN = ALOG10(ZMIN)
     33           30 SC = INC*2. - 1.E-5)/(ZMAX - ZMIN)
     34              DX = ANX/INAX
     35              DY = ANY/JMAX
     36              Y = ANY + DY
     37              DO 80 J = JMAX, 1, -1
     38              Y = Y - DY
     39              N = Y
     40              IF (N .LE. 0) N = 1
     41              IF (N .GE. NY) N = NY - 1
     42              DN = Y - N
     43              X = 0.
     44              DO 70 I = 1, INAX
     45              X = X + DX
     46              IF (I*J .EQ. 1 .OR. I*J .EQ. INAX*JMAX) GO TO 60
     47              IF (I .EQ. 1 .AND. J .EQ. JMAX .OR. J .EQ. 1 .AND.
     48             I I .EQ. INAX) GO TO 60
     49              M = X
     50              IF (M .LE. 0) M = 1
     51              IF (M .GE. NX) M = NX - 1
     52              DM = X - M
     53              IF (LOGFL .NE. 0) GO TO 40
```

```
            C = DN*(DM*Z(N+1,M+1) + (1. - DM)*Z(N+1,M)) + (1. - DN)*
           1 (DM*Z(N,M+1) + (1. - DM)*Z(N,M))
            GO TO 50
         40 C = DN*(DM*ALOG10(Z(N+1,M+1)) + (1. - DM)*ALOG10(Z(N+1,M))) +
           1 (1. - DN)*(DM*ALOG10(Z(N,M+1)) + (1. - DM)*ALOG10(Z(N,M)))
         50 IND = 1 + SC*(C - ZMIN)
            IF (IND .LE. 0) IND = 1
            L(I) = LQ(IND)
            GO TO 70
         60 L(I) = ISTAR
         70 CONTINUE
         80 PRINT 6080, (L(I),I=1,IMAX)
       6080 FORMAT (1X, 132A1)
            IF (LOGFL .EQ. 0) GO TO 90
            ZMAX = HZMAX
            ZMIN = HZMIN
         90 PRINT 6090, ZMIN, ZMAX
       6090 FORMAT ('0MINIMUM VALUE = ', 1PE9.3, 5X, 'MAXIMUM VALUE = ', E9.3)
      C
            RETURN
            END
```

```
S-CUBE0*TPFS.POISPK
      1              SUBROUTINE POISPK
      2        C
      3        C     **THIS SUBROUTINE CONTROLS THE FAST FOURIER SOLUTION OF POISSON
      4        C
      5              DIMENSION U(1),V(1),T(1),R(1),Q(1),SL(1)
      6              REAL PI /3.14159265/
      7        C     **MAKE INDICES GLOBAL
      8              INTEGER NF,NMF,NMH,NM,NH,NHI,NQ,N2,MF
      9              REAL SQUINV
     10        C*********************************************************************
     11        C                                 ENTRY SETUP(NO,MO,Q,R,S,T,SL,U,V)
     12        C*********************************************************************
     13              ENTRY SETUP(NO,MO,Q,R,S,T,SL,U,V)
     14              CALL SET
     15              RETURN
     16        C*********************************************************************
     17        C                                 ENTRY XYPOIS(Q)
     18        C*********************************************************************
     19              ENTRY XYPOIS(Q)
     20        C     **FAST FOURIER ANALYZER
     21              CALL FFANL
     22        C     **VERTICAL SOLUTION BY GAUSS ELIMINATION
     23              CALL GAUSS
     24        C     **FAST FOURIER SYNTHESIZER
     25              CALL FFSYN
     26        C*********************************************************************
     27        C                                 SUBROUTINE SET
     28        C*********************************************************************
     29              SUBROUTINE SET
     30              NH=NO+MO
     31              N=NO+1
     32              NMH=NM/2
     33              NM2=NM+2
     34              NF=NO-1
     35              NMF=NM-NO
     36              NH=NO/2
     37              NHI=NH+1
     38              NQ=NH/2
     39              SQUINV=SQRT(1./NO)
     40              N2=NO+2
     41              NMF2=NMF+2
     42              MF=MO-1
     43              LO=NQ
     44              T(LO)=1.
     45              TSQ=2.
     46              R(LO)=1.
     47            1 LI=LO
     48              LO=L/2
     49              T(LO)=T(L)/(1.+SQRT(TSQ))
     50              TSQ=T(LO)**2 + 1.
     51              R(LO)=2./TSQ
     52              L=LO
     53            2 L=L+LI
```

```
      IF(L.GT.NQ)GO TO 3
      T(L)=(T(L+LO)+T(L-LO))/(2.*T(LO)*(T(L+LO)-T(L-LO)))
      R(L)=2./(1.+T(L)**2)
      GO TO 2
    3 IF(LO.GT.1) GO TO 1
      S4=4.0*S
      DO 4 L=2,NM
      SL(L)=2.0*S4*SIN((PI*(L-1))/NO)**2
      SL(NC-L+2)=SL(L)
    4 CONTINUE
      SL(NM+1)=2.0*S4
      SL(1)=2.0
      RETURN
C*****************************************************************
C                          SUBROUTINE GAUSS
C*****************************************************************
      SUBROUTINE GAUSS
      DO 50 L=1,NO
      ALPHA=SL(L)
      U(1)=0.0
      V(1)=Q(L)
C     **DEVELOP GAUSS ELIMINATION COEFFICIENTS
      I=NO+L
      DO 25 K=2,MF
      U(K)=1.0/(ALPHA-U(K-1))
      V(K)=(V(K-1)-Q(I))*U(K)
      I=I+NO
   25 CONTINUE
C     **BACK SUBSTITUTE
      I=NMF+L-NO
      DO 30 K=MF,2,-1
      Q(I)=V(K)+U(K)*Q(I+NO)
      I=I-NO
   30 CONTINUE
C
   50 CONTINUE
      RETURN
C*****************************************************************
C                          SUBROUTINE FFANL
C*****************************************************************
      SUBROUTINE FFANL
      JB=-NO
      DO 3 K=1,MO,2
      JA=JB+NO
      JB=MIN(JA+NO,NMF+1)
      CALL COOTUK(Q(JA+1),Q(JB+1))
      DO 1 I=2,NO
      U(I)= Q(JA+I)
      V(I)=Q(JB+I)
    1 CONTINUE
      Q(JB+1) =-Q(JB+1)
      Q(JB+NM+1)=-V(NM+1)
      DO 2 I=2,NM
      Q(JA+1)=U(I)+U(N2-I)
```

```
      Q(JA+N2-I)=V(N2-II-V(II
      Q(JB+II=-V(II-V(N2-I)
      Q(JB+N0-I+2)=U(N2-II-U(II
    2 CONTINUE
C
    3 CONTINUE
      RETURN
C*****************************************************************
C                              SUBROUTINE FFSYN
C*****************************************************************
      SUBROUTINE FFSYN
      JB=-NO
      DO 3 K=I,MO,2
      JA=JB+NO
      JB=MIN(JA+NO,NMF+II
      DO ( I=2,NO
      U(I(=Q(I+JA)
      V(I)=Q(I+JB)
    ( CONTINUE
      Q(JB+()=-Q(JB+I)
      Q(JB+NH+(I=-V(NH+I)
      DO 2 I=2,NH
      Q(JA+II=.5*(U(II-V(N2-I)I
      Q(JA+N2-II=.5*(U(II+V(N2-I))
      Q(JB+II=-.5*(U(N2-II+V(II
      Q(JB+N2-II=.5*(U(N2-II-V(II
    2 CONTINUE
      CALL COOTUK(Q(JA+II,Q(JB+II
C
    3 CONTINUE
      RETURN
C*****************************************************************
C                              SUBROUTINE COOTUK(X,YI
C*****************************************************************
      SUBROUTINE COOTUK(X,YI
C
C     **COOLEY-TUKEY FAST FOURIER TRANSFORM ROUTINE
C               (THIS IS A TRUE 'BLACK BOX')
C
      REAL X(II,Y(I)
      NI=I
 16   NU=NH+NI
      NE=NI
 II   RN=R(NE)
      TN=T(NEI
 17   JO=NMI+NE
      JU=NU+NE
      DO (O J=JO,JU
      I=J-NH
      K=I+NE
      L=K+NI
      U(K(=X(II+X(JI
      RE=X(II-X(J)
      V(K(=Y(I)+Y(J(
```

```
      A1=Y(I(-Y(J(
      PN=RN*(RE-TN*A1(
      U(L)=PN-NE
10    V(L)=TN+PN*A1
      TN=-TN
      NM=-NM
      NE=-NE
      IF(NE.LT.0) GO TO 17
      NE=NE+NI
      IF(NE.LT.NQ( GO TO 11
14    NU=NM+NI
      DO 12 K=NM(,NU
      J=K+NQ
      I=K-NQ
      L=K+NI
      U(K(=X(I(+X(J(
      V(L(=X(I(-X(J(
      V(K(=Y(I)+Y(J(
12    U(L(=Y(J)-Y(I(
15    DO 13 I=(,NI
      J=I+NM
      L=I+N(
      U(I(=X(I(+X(J(
      U(L)=X(I(-X(J(
      V(I(=Y(I(+Y(J(
13    V(L(=Y(I(-Y(J)
      IF(NI.EQ.NM( GO TO 22
      DO 18 I=1,NO
      X(I(=U(I)
18    Y(I(=V(I)
      NI=NI+NI
      IF(NI-NQ) 16,19,15
22    DO 24 I=1,NO
      X(I(=SQUINV*U(I)
24    Y(I(=-SQUINV*V(I)
      RETURN
C*****************************************************************
C*****************************************************************
      END
```

```
S-CUBED*TPFS.PRTTST
     1              SUBROUTINE PRTTST
     2              INCLUDE HIFINC
     3
     4        C
     5        C     **THIS SUBROUTINE CONTROLS PRINTER EDITS, PLOTS, AND TAPE DUMPS
     6        C
     7        C     *****PLOTS
     8              IF(IPLOT .EQ. 0) GO TO 30
     9              IF(MOD(ICYCL,IPLOT) .GT. 0.1) GO TO 30
    10              PLOT=.TRUE.
    11           30 IF(TPLOT .LT. 1.0E-8) GO TO 50
    12              DUM(5)=TIMO+TPLOT
    13              IF(TIME .LT. DUM(5)) GO TO 50
    14              TIMO=DUM(5)
    15              PLOT=.TRUE.
    16        C     *****PRINTER EDITS
    17           50 IF(IPRINT .EQ. 0) GO TO 60
    18              IF(MOD(ICYCL,IPRINT) .GT. 0.1) GO TO 60
    19              PRINTX=.TRUE.
    20           60 IF(TPRINT .LT. 1.0E-8) GO TO 80
    21              DUM(6)=TIMM+TPRINT
    22              IF(TIME .LT. DUM(6)) GO TO 80
    23              TIMM=DUM(6)
    24              PRINTX=.TRUE.
    25        C     *****TAPE DUMPS
    26           80 IF(IDUM .EQ. 0) GO TO 90
    27              IF(MOD(ICYCL,IDUMP) .GT. 0.1) GO TO 90
    28              DUMP=.TRUE.
    29           90 IF(TDUMP .LT. 1.0E-8) GO TO 100
    30              DUM(7)=TIMN+TDUMP
    31              IF(TIME .LT. DUM(7)) GO TO 100
    32              TIMN=DUM(7)
    33              DUMP=.TRUE.
    34        C     **ALWAYS PLOT CYCLE ONE, AND DUMP IF DUMPS HAVE BEEN REQUESTED
    35          100 IF(ICYCL .NE. 1) GO TO 150
    36              PLOT=.TRUE.
    37              IF(SAVE) DUMP=.TRUE.
    38        C
    39          150 RETURN
    40              END
```

```
S-CUBED*TPFS,RTAPE
      1            SUBROUTINE RTAPE
      2            INCLUDE HIFINC
      3            DIMENSION LUM(100)
      4            EQUIVALENCE (LUM(1),DUM(1))
      5      C
      6            NTAPE=9
      7      C     **SAVE RESTARTING DATA
      8            SAVE1=RESTRT
      9            SAVE2=TMAX
     10            SAVE3=CYCMAX
     11            ISAVE1=ISTART
     12      C
     13            WRITE(6,1000) ISTART
     14         25 READ (NTAPE) ICTCLE,TIME
     15            WRITE(6,1010) ICYCLE,TIME
     16            IF(ICYCLE .EQ. ISTART) GO TO 100
     17            READ (NTAPE)
     18            GO TO 25
     19      C
     20      C     **READ BLANK COMMON
     21      C
     22        100 READ (NTAPE)  (DUM(I),I=1,17000)
     23      C
     24      C     **EDIT THE INITIAL DATA IN BLANK COMMON
     25      C
     26            WRITE(6,1450)
     27            WRITE(6,1500) (DUM(I),I=1,100)
     28            WRITE(6,1550)
     29            WRITE(6,1600) (LUM(I),I=1,100)
     30      C     **RESTORE RESTARTING DATA
     31            RESTRT=SAVE1
     32            TMAX=SAVE2
     33            CYCMAX=SAVE3
     34            ISTART=ISAVE1
     35      C
     36            RETURN
     37      C
     38      C     **FORMATS
     39      C
     40       1000 FORMAT(1H1,///,43H**********TAPE RESTART REQUESTED FOR CTCLE ,I5,
     41           1   2X,10H********** ,///)
     42       1010 FORMAT(1X, 6HCTCLE ,I5,2X,14HFOUND, TIME = ,1PE14.6)
     43       1450 FORMAT(//,5X,47H**FIRST 100 VALUES IN BLANK COMMON (1PE14.6) **  )
     44       1500 FORMAT(/,(1X,10(1PE12.5,1X)))
     45       1550 FORMAT(//,5X,42H**FIRST 100 VALUES IN BLANK COMMON (I8) **   )
     46       1600 FORMAT(/,(1X,10(I8,5X)))
     47            END
```

```
S-CUBED•TPFS.TIMSTP
     1            SUBROUTINE TIMSTP
     2            INCLUDE HIFINC
     3      C     ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
     4
     5            IF(DTFIX .LT. 1.0E-10) GO TO 10
     6      C     ••FIXED TIME STEP WAS REQUESTED
     7            DT=DTFIX
     8            RETURN
     9      C
    10      C     ••CALCULATION OF TIME STEP
    11      C
    12         10 DT9=DDT•0.1
    13            VTX=0.
    14            VTY=0.
    15            DO 3 I=1,NX
    16            DO 3 J=1,NY
    17            VTX=AMAX1(ABS(U(I,JJ),VTX)
    18            VTY=AMAX1(ABS(V(I,JJ),VTY)
    19          3 CONTINUE
    20      C
    21      C     ••DT2 IS THE VELOCITY-LIMITED TIME CONSTRAINT
    22      C
    23            DT2=1.0E10
    24            IF(VTY .LT. 1.0E-10) GO TO 20
    25            DT2=0.8•DY/VTY
    26         20 IF(VTX .LT. 1.0E-10) GO TO 30
    27            DT2=AMIN1(DT2,0.8•DX/VTX1
    28         30 DT=AMIN1(DT1,DT2)
    29      C     ••LIMIT TIME STEP INCREASE TO 20 PERCENT EACH CYCLE
    30            IF(DT.GT.DDT) DT=DDT
    31            DDT=DT•1.2
    32      C
    33      C     ••RETURN TO MAIN IF TIME STEP IS SATISFACTORY
    34      C
    35            IF (DT.GT.DT9.AND.DT.LT.1.E•10) RETURN
    36      C
    37      C     ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
    38      C     DIAGNOSTIC MESSAGE
    39            PRINT 4
    40            PRINT 5, TQI, TEMP(J), WQI,DT1,DT2,DT3,DT9
    41          4 FORMAT(1HI,'BAD TIMESTEP',10X,'TQI  TEMP(J         WQI  DT1  DT2
    42           2DT3  DT9')
    43          5 FORMAT(16X,1P7E12.3)
    44            STOP
    45            END
```

```
S-CURED*TPFS,UPDXTE
     1              SUBROUTINE UPDXTE
     2              INCLUDE HIFINC
     3        C     ***************************************************************
     4        C     **STATEMENT FUNCTION DERIV2 (SECOND DERIVATIVE)
     5              DERIV2(X,A,C)=X-2.0*A+C
     6        C     ***************************************************************
     7              DTDY=DT*TDT
     8              DVDT=DT*VDT
     9              DTDX=DT*TDX
    10              DVDX=DT*VDX
    11              DO 11 I=1,NX
    12              DO 11 J=1,NY
    13              W(I,J)=0.0
    14              T(I,J)=0.0
    15           11 CONTINUE
    16        C     ***************************************************************
    17        C
    18        C     **SECOND ORDER SCHEME
    19        C
    20              CALL ORDER2(W)
    21              CALL ORDER2(T)
    22        C     ***************************************************************
    23        C     **BYPASS DIFFUSION CALCULATION?
    24              IF(NODIFF .GT. 0) GO TO 75
    25        C
    26        C     **CALCULATE DIFFUSION
    27        C
    28              DO 50 I=1,NX
    29              DO 60 J=1,NY
    30              IF(I.NE.1) GO TO 51
    31              DVX=DERIV2(W(3,J),W(2,J),W(1,J))
    32              DTX=DERIV2(T(3,J),T(2,J),T(1,J))
    33              GO TO 53
    34           51 IF(I.NE.NX) GO TO 52
    35              DVX=DERIV2(W(NX,J),W(NXM1,J),W(NXM2,J))
    36              DTX=DERIV2(T(NX,J),T(NXM1,J),T(NXM2,J))
    37              GO TO 53
    38           52 DVX=DERIV2(W(I+1,J),W(I,J),W(I-1,J))
    39              DTX=DERIV2(T(I+1,J),T(I,J),T(I-1,J))
    40           53 IF(J.NE.1) GO TO 61
    41              DVY=DERIV2(W(I,3),W(I,2),W(I,1))
    42              DTY=DERIV2(T(I,3),T(I,2),T(I,1))
    43              GO TO 63
    44           61 IF(J.NE.NY) GO TO 62
    45              DVY=DERIV2(W(I,NY),W(I,NYM1),W(I,NYM2))
    46              DTY=DERIV2(T(I,NY),T(I,NYM1),T(I,NYM2))
    47              GO TO 63
    48           62 DVY=DERIV2(W(I,J+1),W(I,J),W(I,J-1))
    49              DTY=DERIV2(T(I,J+1),T(I,J),T(I,J-1))
    50           63 W(I,J) = W(I,J) + DVDY + DVY + DVDX + DVX
    51              T(I,J) = T(I,J) + DTDY + DTY + DTDX + DTX
    52           60 CONTINUE
    53           50 CONTINUE
```

```
C     ********************************************************************
C     **UPDATE TEMPERATURE AND VORTICITY
      DO 82 J=1,NY
      DO 81 I=2,NXM1
      W(I,J)=W(I,J)+WTDI(J)*(T(I+1,J)-T(I-1,J))*DT
   81 CONTINUE
C
      W(1,J)=B(1,J)*W(1,J)-BTDI(J)=(T(2,J)-T(NX,J))*DT
      W(NX,J)=B(NX,J)*W(NX,J)-BTDI(J)*(T(1,J)-T(NXM1,J))*DT
      DO 80 I = 1,NX
      T(I,J)=T(I,J)+TDI(J)*0.5*(V(I,J+1)+V(I,J))*DT
   80 CONTINUE
C
   82 CONTINUE
C
      RETURN
      END
```

```
S-CUBED*TPFS.VELOC
      1              SUBROUTINE VELOC
      2              INCLUDE M(FINC
      3              DO 100 J=1,NYP1
      4              DO 100 I=1,NXP1
      5              IF(J .EQ. NYP1) GO TO 50
      6              U(I,J)=(P(I,J+1)-P(I,J))*RDY
      7           50 IF(I .EQ. NXP1) GO TO 100
      8              V(I,J)=(P(I,J)-P(I+1,J))*RDX
      9          100 CONTINUE
     10              RETURN
     11              END
```

```
S-CUBED.TPFS.MTAPE
      1            SUBROUTINE MTAPE
      2                  INCLUDE MIFINC
      3                  MTAPE=9
      4                  WRITE (MTAPE) ICYCL,TIME
      5                  WRITE (MTAPE) (DUM(I),I=1,170001
      6                  WRITE(6,1000) ICYCL,TIME
      7            C
      8                  RETURN
      9            C
     10            C     **FORMATS
     11           1000 FORMAT(1X,13H******CYCLE = ,I5,2X, 7HTIME = ,1PE14.6,2X,
     12                 1   21HDUMPED ON TAPE. ***** )
     13                  END
```

APPENDIX E

A COMPUTER CODE FOR THE ONE-DIMENSIONAL BOUNDARY LAYER

E.1 INTRODUCTION

The radiative transfer task of the Climatology contract calls for the development of an accurate numerical scheme and application of it to the thermodynamics of the atmosphere and soils. Ultimately, this scheme is to be used to perform calibration calculations of the radiative subroutines of General Circulation Models of the Earth's Climate.

In order to develop a realistic radiative transfer code system it is also necessary to take account of effects which strongly influence the thermodynamic state of the atmosphere. By virtue of its strong influence on long wave radiation it is important to take account of water vapor and cloud moisture in the atmosphere. In addition, the state of the lower atmosphere is strongly influenced by turbulent transfer. Consequently, we have formulated and are testing a 1-D computer code to evaluate changes in the atmosphere resulting from the effects of radiative transfer, Coriolis force, turbulent momentum, heat and moisture transfer, and subsidence.

E.2 FORMULATION

The formulation depends only on the vertical coordinate and corresponds to an atmosphere in which all properties are horizontally homogeneous. The atmosphere is described by

the eastward horizontal velocity component u, the northward velocity component v, the temperature T, and the relative humidity q, each of which may depend on the vertical coordinate z. The atmosphere is assumed to be instantaneously in hydrostatic equilibrium but changes in temperature result in a vertical subsidence velocity w. Pressure gradients in the horizontal directions are also taken into account but they are assumed to depend only on z.

The 1-D description of the atmosphere and the hydrostatic approximation permit the use of a Lagrangian formulation in which the atmospheric pressure is a convenient measure of the mass of the atmosphere above the mass element in question. The altitude above the surface z and the pressure p are related by

$$z = -\frac{1}{g}\int_{p}^{p_o} \frac{dp}{\rho} \quad , \tag{E.1}$$

where the density ρ is to be determined from the equation of state, and p_o is the atmospheric pressure corresponding to the surface $z = 0$.

The equation for the transient boundary layer have been treated by many authors, e.g., Estoque,[E1] Pandolfo,[E2] and Sasamori.[E3] We choose to employ the temperature as dependent variable, rather than potential temperature, because temperature is more closely related to the radiative properties which form the most important aspect of this investigation. Using the pressure independent variable the equations are:

$$\frac{du}{dt} = f(v-v_g) + g^2\rho\frac{\partial}{\partial p}\left(K_u\rho\frac{\partial u}{\partial p}\right) \quad ,$$

$$\frac{dv}{dt} = -f(u-u_g) + g^2\rho\frac{\partial}{\partial p}\left(K_v\rho\frac{\partial v}{\partial p}\right) \quad , \qquad \text{(E.2)}$$

$$\frac{dT}{dt} = \left(\frac{\partial F}{\partial p} - w\right)\Gamma + g^2\rho\frac{\partial}{\partial p}\left(K_T\rho\frac{\partial T}{\partial p}\right) - g\rho\Gamma\frac{\partial K_T}{\partial p} \quad ,$$

$$\frac{dq}{dt} = g^2\rho\frac{\partial}{\partial p}\left(K_q\rho\frac{\partial q}{\partial p}\right) \quad .$$

Quantities appearing in Eq. (E.2) are given by:

QUANTITY	SYMBOL	EQUATION	REMARKS
density	ρ	$= \frac{p}{RT}$	determined by equation of state
vertical velocity	w	$= \frac{dz}{dt}$	Lagrangian derivative of altitude
Coriolis parameter	f	$= 2\,\Omega\sin\phi$	ϕ is latitude; Ω is angular velocity of Earth
Geostrophic wind	u_g	$= -\frac{1}{f\rho}\frac{\partial p}{\partial y}$	y is north-south distance
	v_g	$= \frac{1}{f\rho}\frac{\partial p}{\partial x}$	x is east-west distance
Adiabatic lapse rate	Γ	$= \frac{g}{C_p}$	g is gravitational constant; C_p is specific heat of air at constant pressure
Radiation flux	F		to be determined in radiative subroutine
Turbulent diffusivity	K		to be determined in k-subroutine

The indicated time derivatives in Eq. (E.2) are to be formed at constant pressure; consequently, they are Lagrangian derivatives in that they evaluate changes associated with a particular air mass element. The advection term associated with vertical motion is included in these terms.

E.3 EDDY DIFFUSIVITY

The coefficients of turbulent transfer which appear in Eq. (E.2) has been developed through a combination of theoretical considerations and empirical observations. A number of expressions for these quantities are available representing different weightings of the data and greater or lesser sophistication in incorporating theoretical considerations.

Several of the investigators assume that the four eddy coefficients are equal in the two momentum equations and the temperature and relative humidity equation. Such is the case in the work of Sasamori[E3] who uses the eddy diffusion coefficients developed by Yamamoto and Shimanuki.[E4] The same assumption is made by Estoque, et.al.[E1] who attribute their expression to Blackadar.[E5] In our current work we use the prescriptions of Pandolfo[E2] who has modified the Monin-Obukhov formulae as presented by Kitaigorodsky.[E6] Pandolfo takes account of differences between the coefficients for momentum exchanges and those of heat and moisture. He also imposes limitations on the magnitude of the coefficients corresponding to the case of extreme stability.

The expressions for the exchange coefficients are:

Inversion Conditions (Ri > 0)

$$K_u = K_v = K_T = K_g = k^2(Z+Z_o)^2 \left|\frac{\partial U}{\partial Z}\right| (1+\alpha Ri)^2 \qquad (E.3)$$

Lapse Forced-Convection Conditions $(-0.048 \leq Ri \leq 0)$

$$K_u = K_v = k^2(Z+Z_o)^2 \left|\frac{\partial U}{\partial Z}\right| (1-\alpha Ri)^2 \qquad (E.4)$$

$$K_T = K_q = K_u (1-\alpha Ri)^{-2}$$

Lapse Free-Convection Conditions $(Ri < -0.048)$

$$K_T = K_q = h(Z+Z_o)^2 \left|\frac{g}{T}\left(\frac{\partial T}{\partial z} + \Gamma\right)\right|^{\frac{1}{2}} \qquad (E.5)$$

$$K_u = K_v = K_T \left|\frac{3}{c}\right|^{-1/2} \left|Ri\right|^{-1/6} ,$$

where

$$k = 0.4$$

$$\alpha = -3.0$$

$$c = 3\left(\frac{3}{7}\right)^{1/3} \left(\frac{10}{7}\right)^{2}$$

$$h = (0.4)^2 \left(\frac{3}{c}\right)^{3/2}$$

$$Ri = \frac{g}{T}\left[\frac{\partial T}{\partial z} + \Gamma + 0.61T \frac{\partial q}{\partial z}\right] \Big/ \left|\frac{\partial u}{\partial z}\right|^2 .$$

The above values of K are to be restricted in range in order to avoid unrealistic conditions. According to Pandolfo, the K-values should lie within the following ranges:

$10^4 \leq K \leq 10^7$ cm^2/sec if $z > 100$m

$10^2 \leq K \leq 10^7$ cm^2/sec if $z \leq 100$m .

Computed values falling outside of the above range are replaced by the adjacent bounding values.

Clearly, these exchange coefficients are incapable of taking into account such effects as penetrative convection and advection of turbulence, being formulated on the assumption of steady state conditions. We hope to be able to take account of these effects in the future. The influence of penetrative convection has been estimated by Deardorff[E7] and Estoque.[E8] Dynamic effects of turbulence have been treated in varying degrees of rigor by Pritchett and Gawain,[E9] Harlow, et.al.,[E10] and by Donaldson.[E11]

E.4 DIFFERENCE EQUATIONS

The Eq. (E.2) are solved as a set of coupled difference equations in time and space. The difference formulation must satisfy requirements of accuracy, stability, and computational efficiency. Several considerations affecting accuracy, stability and efficiency are discussed below.

Large shear of the horizontal wind in the lower portion of the atmospheric boundary layer results in rapid transport by turbulence and correspondingly large transient adjustments in response to perturbations of the boundary layer. In order to take account of this turbulent transport in a computationally efficient way we have formulated the equations implicitly; the result is an unconditionally stable numerical integration scheme which permits the time interval to be chosen in accord with accuracy considerations. The alternative explicit formulation imposes the requirement that the time interval satisfy the inequality

$$\Delta t \leq \frac{1}{2K}\left(\frac{p}{g\rho}\right)^2 \quad .$$

In the lower boundary layer, the time interval permitted by the above expression will be very small; the diffusivity K is large and the desired pressure interval Δp will be small. The implicit formulation which we have selected (discussed below) requires some additional calculations to solve the sets of simultaneous linear equations. However, the result is a system of equations which are stable for very large time intervals. Time intervals are determined almost entirely by considerations of accuracy.

The changes in the wind, temperature and relative humidity are concentrated predominantly in the lower layers of the boundary layer. From the standpoint of the accuracy and efficiency of the numerical integration it is very desirable to introduce more zones in the region of rapid change near the ground than higher in the atmosphere where much smaller changes occur. In order to achieve spatially variable resolution with accuracy it is necessary to consider carefully the difference formulation. In the following scheme the difference equations retain second order accuracy in regions of variable spatial intervals. The resulting system of equations is capable of representing the atmosphere accurately through the use of a finely resolved layer in the lower boundary layer and increasingly coarse resolution in the higher atmosphere.

We now consider the difference equations corresponding to Eqs. (E.2). The equations are to be solved for the primary dependent variables, u , v , T , q on a discrete mesh on both of the indpendent variables, pressure p and time t . The resulting difference equations are to be solved by marching the solution forward by successive increments Δt of the time. In order to use a closed-form soiution of the implicit system, we linearize those terms of the equations which depend on the advanced time.

We denote a discrete value of the time by superscript n, i.e., $t^{n+1} = t^n + \Delta t^n$. The atmospheric pressure is partitioned into intervals Δp_i such that $p_{i-\frac{1}{2}} + \Delta p_i = p_{i+\frac{1}{2}}$. The difference equations corresponding to Eqs. (E.2) are:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t^n} = f\left(v_i^{n+1} - v_{g,i}\right) + \sigma_i^n\left[(K_u\rho)^n_{i+\frac{1}{2}} \frac{u_{i+1}^{n+1} - u_i^n}{\Delta p_{i+1} + \Delta p_i} - (K_u\rho)^n_{i-\frac{1}{2}} \frac{u_i^{n+1} - u_{i-1}^{n+1}}{\Delta p_i + \Delta p_{i-1}}\right], \qquad \text{(E.6)}$$

$$\frac{v_i^{n+1} - v_i^n}{\Delta t^n} = f\left(u_{g,i} - u_i^{n+1}\right) + \sigma_i^n\left[(K_v\rho)^n_{i+\frac{1}{2}} \frac{v_{i+1}^{n+1} - v_i^{n+1}}{\Delta p_{i+1} + \Delta p_i} - (K_v\rho)^n_{i-\frac{1}{2}} \frac{v_i^{n+1} - v_{i-1}^{n+1}}{\Delta p_i + \Delta p_{i-1}}\right], \qquad \text{(E.7)}$$

$$\frac{T_i^{n+1} - T_i^n}{\Delta t^n} = \Gamma\left(\frac{F^n_{i+\frac{1}{2}} - F^n_{i-\frac{1}{2}}}{\Delta p_i} - w_i^n\right) - \frac{\sigma_i^n}{2g}\Gamma\left(K^n_{T,i+\frac{1}{2}} - K^n_{T,i-\frac{1}{2}}\right) + \sigma_i^n\left[(K_T\rho)^n_{i+\frac{1}{2}} \frac{T_{i+1}^{n+1} - T_i^{n+1}}{\Delta p_{i+1} + \Delta p_i} - (K_T\rho)^n_{i-\frac{1}{2}} \frac{T_i^{n+1} - T_{i-1}^{n+1}}{\Delta p_i + \Delta p_{i-1}}\right].$$

$$\frac{q_i^{n+1} - q_i^n}{\Delta t^n} = \sigma_i^n \left[(K_q \rho)^n_{i+\frac{1}{2}} \frac{q_{i+1}^{n+1} - q_i^{n+1}}{\Delta p_{i+1} + \Delta p_i} \right.$$

$$\left. - (K_q \rho)^n_{i-\frac{1}{2}} \frac{q_i^{n+1} - q_{i-1}^{n+1}}{\Delta p_i + \Delta p_{i-1}} \right] . \qquad \text{(E.9)}$$

In the above equations we have defined

$$\sigma_i^n = \frac{2g^2 \rho_i^n}{\Delta p_i} ,$$

and the diffusion coefficients are to be evaluated from Eqs. (E.3) through (E.5) using appropriate centered difference representations of the dependent variables from cycle n.

The density is obtained from the equation of state as follows:

$$\rho_i^{n+1} = \frac{p_i}{R T_i^{n+1}} ,$$

where

$$p_i = \tfrac{1}{2}(p_{i+\frac{1}{2}} + p_{i-\frac{1}{2}}) .$$

The altitude corresponding to the pressure also depends on time by virtue of the hydrostatic readjustment of the vertical column under the changing temperature:

$$z_{i+\frac{1}{2}}^{n+1} = \frac{1}{g} \sum_{k=i}^{I} \frac{\Delta p_k}{\rho_k^{n+1}} \quad ,$$

where I is the maximum value of i corresponding to the zone adjacent to the ground.

The vertical velocity is obtained as a difference

$$w_i^{n+1} = \frac{z_i^{n+1} - z_i^n}{\Delta t^n} \quad ,$$

where

$$z_i = \tfrac{1}{2}(z_{i+\frac{1}{2}} + z_{i-\frac{1}{2}}) \quad .$$

Considering the above as a system of simultaneous equations for the unknown quantities u_i^{n+1}, v_i^{n+1}, T_i^{n+1}, g_i^{n+1}, we note that the equations are linear and are uncoupled in the following way: The T-equation and q-equation are not coupled to each other or to the u or v equations; the u and v equations are coupled together through the Coriolis terms. Consequently, the T and q equations can each be represented as a tridiagonal equation for a scalar quantity. The u and v equations, however, are conveniently represented together as a tridiagonal system of equations for a vector quantity having the two components u_i and v_i.

The scalar equations can be represented in the form

$$A_i \phi_{i+1} + B_i \phi_i + C_i \phi_{i-1} = D_i \quad . \qquad \text{(E.10)}$$

For the T-equation,

$$\phi_i = T_i^{n+1}$$

and

$$A_i = - \frac{\sigma_i^n \, \Delta t (K_T \rho)_{i+\frac{1}{2}}^n}{\Delta p_{i+1} + \Delta p_i} \, ,$$

$$B_i = 1 - A_i - C_i \, ,$$

$$C_i = - \frac{\sigma_i^n \, \Delta t (K_T \rho)_{i-\frac{1}{2}}^n}{\Delta p_i + \Delta p_{i-1}} \, ,$$

$$D_i = \Gamma \, \Delta t \left[\frac{F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n}{\Delta p_i} - w_i^n - \frac{\sigma_i^n}{2g} \left(K_{T,i+\frac{1}{2}} - K_{T,i-\frac{1}{2}} \right) \right] + T_i^n \, .$$

For the q-equation,

$$\phi_i = q_i^{n+1}$$

and

$$A_i = - \frac{\sigma_i^n \, \Delta t (K_q \rho)_{i+\frac{1}{2}}^n}{\Delta p_{i+1} + \Delta p_i} \, ,$$

$$B_i = 1 - A_i - C_i \quad ,$$

$$C_i = - \frac{\sigma_i^n \, \Delta t (K_q \rho)_{i-\frac{1}{2}}^n}{\Delta p_i + \Delta p_{i-1}} \quad ,$$

$$D_i = q_i^n \quad ;$$

The vector equation can also be represented in the form of Eq. (E.1) where

$$\phi_i = \begin{pmatrix} u_i^{n+1} \\ v_i^{n+1} \end{pmatrix}$$

is a vector quantity and the coefficients are matrices having the form

$$A_i = \begin{pmatrix} -U^+ & 0 \\ 0 & -V^+ \end{pmatrix} \quad ,$$

$$B_i = I - A_i - C_i + \begin{pmatrix} 0 & -f\Delta t \\ f\Delta t & 0 \end{pmatrix}$$

$$C_i = \begin{pmatrix} -U^- & 0 \\ 0 & -V^- \end{pmatrix}$$

$$D_i = \begin{pmatrix} u_i^n - \Delta t f\, v_{g,i} \\ v_i^n + \Delta t f\, u_{g,i} \end{pmatrix} .$$

where

$$U^+ = \frac{\sigma_i^n\, \Delta t (K_u \rho)^n_{i+\frac{1}{2}}}{\Delta p_i + \Delta p_{i+1}} \quad , \quad V^+ = \frac{\sigma_i^n\, \Delta t (K_v \rho)^n_{i+\frac{1}{2}}}{\Delta p_{i+1} + \Delta p_i} \quad ,$$

$$U^- = \frac{\sigma_i^n\, \Delta t (K_u \rho)^n_{i-\frac{1}{2}}}{\Delta p_i + \Delta p_{i-1}} \quad , \quad V^- = \frac{\sigma_i^n\, \Delta t (K_v \rho)^n_{i-\frac{1}{2}}}{\Delta p_i + \Delta p_{i-1}} \quad ,$$

and I is the identity matrix.

All of these systems of equations can be solved readily by Gaussian elimination. Taking advantage of the tri-diagonal form of the equations, the solution algorithm reduces to evaluation of coefficients recursively in one forward and one backward sweep through the mesh. The algorithm for vector equations is discussed by Richtmyer and Morton.(E12)

E.5 BOUNDARY VALUES

The calculational region extends from the ground, where the pressure has the assumed ground level hydrostatic value, to

an arbitrary altitude having a specified pressure. Boundary conditions are required at the top and bottom of the mesh to close the system of equations. We have not investigated these conditions carefully (they will be affected further by the radiative treatment at the ground), but are using the following set: At the ground level the velocity is zero, corresponding to the viscous boundary condition, the temperature has a specified value, and the relative humidity is given the saturation value corresponding to the ground temperature. At the top of the mesh the velocity takes the geostrophic value and temperature and humidity are specified.

REFERENCES

E1 Estoque, M. and C. Bhumralkar, "A Method for Solving the Planetary Boundary-Layer Equations," Boundary Layer Meteorol. 1, 169 (1970).

E2 Pandolfo, J., "A Numerical Model of the Atmosphere -- Ocean Planetary Boundary Layer," Proceedings, WMO/IUGG Symposium on Numerical Weather Prediction, Tokyo, November 1969.

E3 Sasamori, T., "A Numerical Study of Atmospheric and Soil Boundary Layers," J. Atmos. Sci. 27, 1122 (1970).

E4 Yamamoto, G. and A. Shimanuki, "Turbulent Transfer in Diabatic Conditions," J. Meteorl. Soc. Japan, Ser. 2, 44, 301 (1966).

E5 Blackadar, A., "The Vertical Distribution of Wind and Turbulent Exchange in a Neutral Atmosphere," J. Geophys. Res. 67, 3095 (1962).

E6 Kitaigorodsky, S., "On the Possibility of Theoretical Calculation of Vertical Temperature Profile in Upper Layer of the Sea," Bull. Acad. Sci., U.S.S.R., Geophys. Ser. 3, 313 (1961).

E7 Deardorff, J., "The Counter-Gradient Heat Flux in the Lower Atmosphere and in the Laboratory," J. Atmos. Sci. 25, 503 (1966).

E8 Estoque, M., "Vertical Mixing Due to Penetrative Convection," J. Atmos. Sci. 25, 1046 (1968).

E9 Gawain, T. and J. Pritchett, "A Unified Heuristic Model of Fluid Turbulence," J. Comp. Phys. 5, 383 (1970).

E10 Harlow, F. and P. Nakayama, "Turbulence Transport Equations," Phys. Fluids 10, 2323 (1967).

E11 Donaldson, C., "Calculation of Turbulent Shear Flows for Atmospheric and Vortex Motions," AIAA Paper No. 71-217, January 1971.

E12 Richtmyer, R. and K. Morton, "Difference Methods for Initial-Value Problems, Interscience Publications, New York, pp. 275 (1967).